Pervasive.SQL V8

User's Guide

Guide to Using Pervasive.SQL

Pervasive Software Inc.
12365 Riata Trace Parkway
Building B
Austin, TX 78727 USA

Telephone: 512 231 6000 or 800 287 4383
Fax: 512 231 6010
Email: info@pervasive.com
Web: http://www.pervasive.com

disclaimer

PERVASIVE SOFTWARE INC. LICENSES THE SOFTWARE AND DOCUMENTATION PRODUCT TO YOU OR YOUR COMPANY SOLELY ON AN "AS IS" BASIS AND SOLELY IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF THE ACCOMPANYING LICENSE AGREEMENT. PERVASIVE SOFTWARE INC. MAKES NO OTHER WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, REGARDING THE SOFTWARE OR THE CONTENT OF THE DOCUMENTATION; PERVASIVE SOFTWARE INC. HEREBY EXPRESSLY STATES AND YOU OR YOUR COMPANY ACKNOWLEDGES THAT PERVASIVE SOFTWARE INC. DOES NOT MAKE ANY WARRANTIES, INCLUDING, FOR EXAMPLE, WITH RESPECT TO MERCHANTABILITY, TITLE, OR FITNESS FOR ANY PARTICULAR PURPOSE OR ARISING FROM COURSE OF DEALING OR USAGE OF TRADE, AMONG OTHERS.

trademarks

Btrieve, Client/Server in a Box, Pervasive, Pervasive Software, and the Pervasive Software logo are registered trademarks of Pervasive Software Inc.

Built on Pervasive Software, DataExchange, MicroKernel Database Engine, MicroKernel Database Architecture, Pervasive.SQL, Solution Network, Ultralight, and ZDBA are trademarks of Pervasive Software Inc.

Microsoft, MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows Millennium, Windows 2000, Windows XP, Win32, Win32s, and Visual Basic are registered trademarks of Microsoft Corporation.

NetWare and Novell are registered trademarks of Novell, Inc.

NetWare Loadable Module, NLM, Novell DOS, Transaction Tracking System, and TTS are trademarks of Novell, Inc.

All other company and product names are the trademarks or registered trademarks of their respective companies.

© Copyright 2002 Pervasive Software Inc. All rights reserved. Reproduction, photocopying, or transmittal of this publication, or portions of this publication, is prohibited without the express prior written consent of the publisher.

This product includes software developed by Powerdog Industries. © Copyright 1994 Powerdog Industries. All rights reserved. This product includes software developed by KeyWorks Software. © Copyright 2002 KeyWorks Software. All rights reserved. This product includes software developed by DUNDAS SOFTWARE. © Copyright 1997-2000 DUNDAS SOFTWARE LTD., all rights reserved. This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

The ODBC Driver Manager for NetWare (ODBC.NLM) included in this product is based on the GNU iODBC software © Copyright 1995 by Ke Jin <kejin@empress.com> and was modified by Simba Technologies Inc. in June 1999.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

A copy of the GNU Lesser General Public License is included in your installation of Pervasive.SQL at \program files\common files\Pervasive Software Shared\doc\lesser.htm. If you cannot find this license, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA. You may contact Pervasive Software Inc. using the contact information on the back cover of this manual.

User's Guide

November 2002

100-004151-001

Contents

About This Manual . . . . . **xiii**

Who Should Read This Manual xiv
Manual Organization xv
Conventions xvii

1 Introducing Pervasive.SQL **1-1**

Understanding Pervasive.SQL and its Capabilities

Understanding Pervasive.SQL 1-2
What is a Database? 1-2
What is a Database Management System? 1-2
Components of Pervasive.SQL 1-3
Key Concepts 1-5
Basic Database Structures and Terms 1-5
Unique Benefits of Pervasive.SQL 1-8
Why Pervasive.SQL? 1-10
Understanding the Product Family 1-12
Workgroup 1-12
Server 1-13
DataExchange 1-14
Crystal Reports for Pervasive.SQL 1-14
Helpful Utilities 1-15
Using Pervasive.SQL Documentation 1-17
Getting Started Guides 1-17
What's New 1-18
User's Guide 1-18
Advanced Operations Guide 1-18
SQL Engine Reference 1-18
Status Codes and Messages 1-18
Pervasive Products and Services 1-18
Online Help 1-19
File System Security 1-20

2 Using Pervasive.SQL **2-1**

A Walk-through of Basic User Tasks

Starting and Stopping the Database Engine 2-2
Starting and Stopping the Server Engine on a Windows Server 2-3
Starting and Stopping the Workgroup Engine on Windows 2-7
Starting and Stopping the Database Engine on NetWare 2-8
Starting and Stopping the Database Engine on Linux 2-9

Granting Administrative Rights for the Database Engine . . . 2-10
Tasks Requiring Administrative Rights . . . 2-10
How Administrative Rights are Granted . . . 2-10
Rights Within an Active Directory Environment . . . 2-11
Rights Provided to non-Administrative Users . . . 2-11
Tasks for Granting Administrative Rights . . . 2-11
Granting Administrative Rights on a Windows Server . . . 2-12
Granting Administrator Rights on NetWare 4.2, 5.x or 6.x . . . 2-19
Granting Administrator Rights on Linux . . . 2-19
Logging in as Administrator on any platform . . . 2-20
Setting Up ODBC Database Access . . . 2-21
Basic Concepts . . . 2-21
What to Know if You are Recreating DSNs . . . 2-23
Prerequisites to Setting up Database Access . . . 2-24
Setting Up Database Access on Windows . . . 2-25
Setting Up Database Access on a NetWare Server . . . 2-31
Setting Up Database Access on a Linux Server . . . 2-37
Setting Up Client Access . . . 2-40
Accessing Data via ODBC From Other Applications . . . 2-46
Before You Begin . . . 2-46
Accessing Data Using Microsoft Excel . . . 2-46
Accessing Data Using Microsoft Access . . . 2-48
Deleting DSNs . . . 2-52
Bound Databases and Enforced Integrity . . . 2-55
Using the Fast User Switching Feature of Windows XP . . . 2-56

3 Using the Pervasive Control Center . . . 3-1

A Brief Tour of Pervasive Control Center

An Overview of Pervasive Control Center . . . 3-2
Registering or Removing a Server . . . 3-4
Viewing Database Engines . . . 3-6
Interpreting Engine Status Icons . . . 3-6
Pervasive Control Center Wizards . . . 3-8
Adding or Creating a Database . . . 3-10
Deleting a Database . . . 3-13
Adding a Table . . . 3-15
Dropping a Table . . . 3-20
Setting Database Security . . . 3-22
Turning Security On and Off . . . 3-22
Working with Groups and Users . . . 3-23
Stopping and Restarting Services on Windows Servers . . . 3-28
Viewing and Modifying Table Properties . . . 3-29
Viewing and Modifying Data . . . 3-32

Exporting/Importing Data 3-33
Export and Import Procedures 3-33
Checking Consistency and Referential Integrity 3-42
Listing Referential Constraints 3-42
Checking Consistency 3-43
Checking Referential Integrity 3-44

4 License Administrator 4-1

Working with License Keys and User Counts

License Administrator Concepts 4-2
Graphical User Interface 4-2
Command Line Interface 4-2
User Count 4-3
Licenses for Prior Versions of Pervasive.SQL 4-4
License Administrator GUI Visual Reference 4-5
GUI Dialog 4-5
License Administrator Tasks 4-9
GUI Tasks 4-10
CLI Tasks 4-16

5 Table Designer 5-1

A Tour of Pervasive Table Designer

Table Designer Concepts 5-2
Overview 5-2
Tabbed Dialogs 5-3
Modes 5-5
Data Types 5-6
Null Values 5-6
PVS File 5-7
Table Designer GUI Visual Reference 5-8
Main Window 5-8
Columns Tab 5-18
Indexes Tab 5-20
Statistics Tab 5-21
Btrieve Tab 5-25
Table Designer Tasks 5-28
General Tasks 5-28
Columns Tasks 5-29
Indexes Tasks 5-29
Statistics Tasks 5-30
Btrieve Tasks 5-30

6 SQL Data Manager 6-1

A Tour of Pervasive SQL Data Manager

SQL Data Manager Concepts 6-2
Overview 6-2
Restrictions 6-3
Results Tabs 6-4
Panes 6-8
SQL Reference 6-15
SQL Data Manager GUI Visual Reference 6-16
Main Window 6-16
Query Builder Diagram 6-31
Query Builder Grid for SELECT Statements 6-32
Query Builder Grid for UPDATE Statements 6-35
Query Builder Grid for INSERT Statements 6-36
Query Builder Grid for DELETE Statements 6-37
SQL Data Manager Tasks 6-40
General Tasks 6-40
Query Pane Tasks 6-40
Execution and Results Tasks 6-41
Query Builder Tasks. 6-41

7 Pervasive System Analyzer (PSA) 7-1

Usage Help for the Diagnostic Utility in Pervasive.SQL V8

PSA Concepts. 7-2
PSA Functional Overview. 7-2
Why Use PSA? 7-3
PSA Tests in Detail 7-3
Frequently Asked Questions 7-7
Introductory Network Information 7-11
PSA GUI Visual Reference. 7-12
PSA Tasks 7-15
Analyze and Archive Tasks 7-16
Restore Archive Tasks. 7-17
Delete Components or Archives Tasks 7-18
View Components Tasks 7-21
Test Active Installation Tasks 7-24
Other Tasks 7-28

8 Linux Supplementary Documentation 8-1

Additional Information on Linux Utilities

User Manual Exclusions for Linux 8-2
Utilities 8-2
Changes and Exclusions 8-2
Man Pages 8-4
Available Utilities 8-5

9 Basic Troubleshooting 9-1

How to Identify and Solve Common Problems

General Troubleshooting 9-2
Error Messages from PCC 9-8
Frequently Asked Questions 9-12
Installation 9-15
Security 9-16
Documentation 9-17
User Counts 9-18
Networking 9-18
Difficulty Accessing Data 9-19
ODBC and DDFs 9-21
Upgrading from Btrieve 6.15 9-27
Upgrading and Migration 9-28
Miscellaneous 9-29

10 Pervasive.SQL Resources and Contacts 10-1

A Guide to Pervasive.SQL Customer Information Resources

Printed Documentation 10-2
Developer Center 10-3
Pervasive.SQL Knowledge Base 10-4
FTP Site 10-5
Online Documentation 10-6
Pervasive Library 10-6
Webinars 10-7
Subscription Based E-mail Services 10-8
DevTalk 10-9
Newsgroup 10-10
E-Mail Contacts 10-11
Technical Support 10-12

Figures

2-1 Windows NT Services Dialog Box 2-3
2-2 Windows 2000 Services Dialog Box 2-4
2-3 Windows NT Services Dialog Box 2-5
2-4 Windows 2000 Services Dialog Box 2-5
2-5 Connect to Remote Server Dialog Box 2-20
2-6 Example DSN Configurations 2-22
2-7 Registering a New Engine. 2-25
2-8 Choosing a Computer Name. 2-25
2-9 Choosing a Computer Name from a Network List 2-26
2-10 Create an Engine DSN and Connect to a Server Database. 2-27
2-11 Naming the New Database on the Server 2-27
2-12 Advanced Settings Window 2-28
2-13 Create DBNAME Window 2-29
2-14 Registering a New Engine. 2-31
2-15 Choosing a Computer Name. 2-32
2-16 Choosing a Computer Name from a Network List 2-32
2-17 Create an Engine DSN and Connect to a Server Database. 2-33
2-18 Naming the New Database on the Server 2-33
2-19 Create Database Wizard Complete 2-34
2-20 Advanced Settings Window 2-35
2-21 Create DBNAME Window 2-35
2-22 Registering a New Engine. 2-40
2-23 Choosing a Computer Name. 2-41
2-24 Choosing a Computer Name from a Network List 2-41
2-25 Create Database Wizard—Client DSN 2-41
2-26 Create Database Wizard—Client DSN Confirmation 2-42
2-27 Pervasive ODBC Client DSN Setup Screen 2-43
2-28 Pervasive ODBC Client DSN Options. 2-44
2-29 Pervasive ODBC Client DSN Setup Screen #2 2-44
2-30 Accessing Pervasive Data using Microsoft Excel 2-47
2-31 Excel Display of ODBC Source List 2-47
2-32 Create a New Database using Microsoft Access. 2-48
2-33 Importing External Data Using Access 2-49
2-34 Access Display of ODBC Source List 2-49
2-35 Using Pervasive Data in Microsoft Access. 2-51

3-1 Pervasive Control Center 3-2
3-2 Registering a New Engine 3-4
3-3 Choosing a Computer Name 3-4
3-4 Choosing a Computer Name from a Network List 3-5
3-5 DOS Command Line Screen Example 3-8
3-6 Create Database Wizard Dialog Box 3-10
3-7 Create Database Wizard - Complete Dialog Box 3-12
3-8 Drop Database Wizard Dialog Box 3-13
3-9 Completing Drop Database Wizard Dialog Box. 3-14
3-10 Create Table Wizard Dialog Box 3-15
3-11 Create Table Wizard - Designer View Dialog Box. 3-16
3-12 Create Table Wizard - Generate SQL Script Dialog Box 3-18
3-13 Create Table Wizard - Complete Dialog Box 3-19
3-14 Drop Table Wizard. 3-20
3-15 Database Properties Dialog Box. 3-22
3-16 Add New Group Dialog Box 3-24
3-17 Add New User / Group Permissions Dialog Box 3-24
3-18 Add New User Dialog Box. 3-25
3-19 Add New User / Group Permissions Dialog Box 3-26
3-20 Stopping and Restarting Services in the Pervasive Control Center. 3-28
3-21 Table Properties Window 3-29
3-22 Export Wizard - Choose a Destination File Dialog Box. 3-34
3-23 Export Wizard - Specify Table Copy or Query Dialog Box 3-34
3-24 Export Wizard - Use SQL Statement for Export Dialog Box 3-35
3-25 Export Wizard - Completing the Export to the File Wizard Dialog Box 3-35
3-26 Exporting Data Status 3-36
3-27 Import Wizard - Choose a Source File Dialog Box 3-36
3-28 Import Wizard - Specify Table Copy Dialog Box 3-37
3-29 Import Wizard - Use SQL Statement for Import Dialog Box. 3-37
3-30 Import Wizard - Completing the Import from File Wizard Dialog Box. 3-38
3-31 Importing Data Dialog Box 3-39
3-32 Database Properties 3-42
3-33 Check Database Dialog Box 3-43
3-34 Select Tables for the Referential Integrity Test Dialog Box 3-43
3-35 Check Database Dialog Box 3-44
3-36 Details of the Referential Integrity Test Dialog Box 3-44
3-37 Select Tables for the Referential Integrity Test Dialog Box 3-45
3-38 Check Database Results Dialog Box 3-45
3-39 Referential Integrity Test Results Dialog Box 3-46

5-1 Tabbed Dialogs in Table Designer 5-3
5-2 Mode Indicator 5-5
5-3 Table Designer Main Window 5-9
5-4 Columns Tab 5-19
5-5 Indexes Tab 5-20
5-6 Statistics Tab 5-22
5-7 Btrieve Tab 5-25

6-1 Primary Areas of SQL DM Graphical User Interface 6-3
6-2 SQL DM Results Tabs 6-4
6-3 Grid Results Tabbed Dialog 6-5
6-4 Scroll Box Tool Tip Indicating Row at Top of Grid 6-7
6-5 Text Result Tab View 6-8
6-6 Query Builder Diagram with Tables Class and Faculty 6-11
6-7 Results of Showing Query Builder Diagram Containing No Tables 6-13
6-8 Example of Join Line Between Columns 6-14
6-9 SQL Data Manager Main Window 6-17
6-10 Query Builder Grid for SELECT Statements 6-32
6-11 Query Builder Grid for UPDATE Statements 6-35
6-12 Query Builder Grid for INSERT Statements 6-37
6-13 Query Builder Grid for DELETE Statement 6-38

7-1 PSA Main Dialog 7-12
7-2 View Modules Section of PSA 7-21
7-3 View Modules Results 7-23
7-4 Multiple Occurrences of a Component Found 7-24
7-5 Sorting Multiple Occurrences of a Component 7-24
7-6 Selecting the Machine on which to Run a Network Test 7-25
7-7 Selecting Protocols for Network Tests 7-26
7-8 Selecting Number of Stress Test Messages 7-26
7-9 Successful Network Test Screen 7-26
7-10 Specifying the PSA Log File Name and Location 7-28
7-11 Searching for PSA Log File Path 7-29
7-12 Viewing the PSA Log File 7-29
7-13 Viewing the PSA Version 7-30

Tables

1-1 Comparison of Server and Workgroup Features 1-13
1-2 Summary of Pervasive.SQL Utilities 1-15

3-1 Pervasive.SQL Engine States. 3-7
3-2 File Names of Wizards 3-8
3-3 Table Wizard Tools 3-16
3-4 Existing Table Properties 3-30
3-5 Import/Export Wizards Data Formats 3-33

4-1 License Administrator Command Line Options and Parameters 4-3

5-1 Comparison of File Menu Commands 5-3

6-1 Comparison of Active and Display-only Grid 6-5
6-2 Icons in Results Grid Indicator Column 6-6
6-3 Symbols Used Within a Table Window 6-11

7-1 Components Grid Items. 7-23

About This Manual

This manual introduces you to Pervasive.SQL utilities for Server and Workgroup products and shows you how to perform the basic tasks necessary to work with the application. Topics include starting and stopping the database engine, setting up access to a database, and accessing data from other applications. This manual also gives you a tour of the Pervasive Control Center (PCC). PCC allows you to manage Pervasive.SQL utilities within a single, easy-to-use framework.

Who Should Read This Manual

This manual provides information for users who install and run Pervasive.SQL client/server and Workgroup products.

Pervasive Software would appreciate your comments and suggestions about this manual. Please send comments to docs@pervasive.com.

Manual Organization

This manual is divided into the following chapters:

- Chapter 1—"Introducing Pervasive.SQL"

  This chapter provides an introduction to Pervasive.SQL, and an overview of utilities and Pervasive.SQL documentation.

- Chapter 2—"Using Pervasive.SQL"

  This chapter covers the basic tasks required to work with Pervasive.SQL.

- Chapter 3—"Using the Pervasive Control Center"

  This chapter explains how to get your work done using Pervasive Control Center.

- Chapter 4—"License Administrator"

  This chapter explains how to increase user counts and add, delete, and view license keys.

- Chapter 5—"Table Designer"

  This chapter explains how to add, change, or delete databases and tables using the Table Designer.

- Chapter 6 — "SQL Data Manager"

  This chapter explains how to use the SQL Data Manager to run SQL statements against a Pervasive.SQL database.

- Chapter 7—"Pervasive System Analyzer (PSA)"

  This chapter explains how to work with archived Pervasive.SQL components and test an active installation.

- Chapter 8—"Linux Supplementary Documentation"

  This chapter explains how to get things done on Linux server platforms.

- Chapter 9—"Basic Troubleshooting"

  This chapter provides information for troubleshooting and resolving problems.

- Chapter 10—"Pervasive.SQL Resources and Contacts"

  This chapter explains the resources and information at your disposal as a valued customer of Pervasive Software.

The manual also includes an index.

Conventions

Unless otherwise noted, command syntax, code, and examples use the following conventions:

CASE	Commands and reserved words typically appear in uppercase letters. Unless you are working with Linux or the manual states otherwise, you can enter these items using uppercase, lowercase, or both. For example, you can type `MYPROG`, `myprog`, or `MYprog`.
Bold	Words appearing in bold include the following: menu names, dialog box names, commands, options, buttons, statements, etc.
`Monospaced font`	Monospaced font is reserved for words you enter, such as command syntax.
[ ]	Square brackets enclose optional information, as in [*log_name*]. If information is not enclosed in square brackets, it is required.
|	A vertical bar indicates a choice of information to enter, as in [*file name* | *@file name*].
< >	Angle brackets enclose multiple choices for a required item, as in `/D=<5|6|7>`.
variable	Words appearing in italics are variables that you must replace with appropriate values, as in *file name*.
…	An ellipsis following information indicates you can repeat the information more than one time, as in [*parameter* …].
::=	The symbol ::= means one item is defined in terms of another. For example, a::=b means the item *a* is defined in terms of *b*.

chapter

1 Introducing Pervasive.SQL

Understanding Pervasive.SQL and its Capabilities

This chapter provides an explanation of what Pervasive.SQL is and what it can do for you. This chapter is divided into the following sections:

- "Understanding Pervasive.SQL" on page 1-2
- "Key Concepts" on page 1-5
- "Why Pervasive.SQL?" on page 1-10
- "Understanding the Product Family" on page 1-12
- "Helpful Utilities" on page 1-15
- "Using Pervasive.SQL Documentation" on page 1-17
- "File System Security" on page 1-20

Understanding Pervasive.SQL

Pervasive.SQL is a comprehensive database management system built around Pervasive Software's MicroKernel Database Engine. Pervasive.SQL offers easy installation, uncomplicated maintenance, and high levels of performance and reliability.

This section explains the product and its components.

What is a Database?

Loosely defined, a *database* is simply a collection of data. Generally, the data is structured by dividing it into sub-sets of information that share the same characteristics. Some examples of a database are:

- A telephone book

  Each entry in the phone book consists of four characteristics: first name, last name, address, and phone number.

- A collection of digital photographs

  Each picture on your hard disk has two characteristics: a file name, and the data within the file that represents the image.

- A list of orchards and the fruit grown by each

  Each entry in the orchard list might consist of three characteristics: orchard name, address, and date founded. The related list of fruits might have five characteristics: orchard name, fruit name, fruit type (McIntosh, Fuji, and so on), fruit price, and a taste rating.

In the particular context of this product, a database is a specific, well-defined collection of related information. You can probably find one or more databases available on your computer or your network. For example, you may have a database of information related to vendors from whom you purchase supplies or raw materials, and you probably also have a database containing customer or member information. Each of these is a distinct, well-defined collection of related information.

What is a Database Management System?

As citizens of the computer age, we are surrounded by collections of information—databases—everywhere we go. Unfortunately, all this data is of no use to anyone without methods to sort it, search it, analyze it, and keep it current.

A *database management system, or DBMS,* is a computer program designed to manage large amounts of data and to allow other computer programs and people to interact with the data. A DBMS can also be referred to informally as a *database engine* or simply an *engine*. A DBMS performs the following tasks:

- *Controls access to the data.* The DBMS can act as a watchdog to prevent the wrong people from using the data.
- *Structures the data so it can be interpreted by other applications.* The DBMS ensures that all the data adheres to the database structure, so that other computer programs can work with the data using common methods.
- *Keeps the data safe and prevents it from getting garbled or lost.* The DBMS facilitates backing up the data in case of catastrophic loss, and also accesses it in a consistent manner to prevent the data from suffering inadvertent damage.
- *Makes it easy to add new information, find it, update it, and delete it.* The DBMS readily accepts new data and provides tools that you can use to locate, update, and remove information as you see fit. It verifies that the data inserted fits within defined attributes for the database fields.
- *Allows you to analyze relationships among different sets of data.* The DBMS stores the data in a way that allows you to examine how any piece of data relates to any other piece of data.

In summary, the DBMS organizes your data, keeps it safe, and helps you to use it and understand it.

Components of Pervasive.SQL

The Pervasive.SQL DBMS consists of a variety of components designed to help you achieve your data management goals. For more detailed information about these components and how they interact, please see *Pervasive Products and Services*.

MicroKernel Database Engine

The MicroKernel Database Engine (MKDE) is the high-performance heart of Pervasive.SQL. The MKDE works directly with the data files on your computer's hard disk. When requested, it directly inserts new data, deletes unnecessary data, and ensures the safety and integrity of the data files at all times, even when people and applications are working with the data.

SQL Relational Database Engine

The SQL Relational Database Engine (SRDE) interacts with the MKDE and the client (described below). It provides many powerful features including support for Microsoft ODBC, sophisticated search and analysis capability, and security.

Client (also called Requester)

In client/server systems, the client resides on the computer workstation. The client interacts with the client application and across the network with both the MKDE and the SRDE on the server.

Pervasive Control Center

The Pervasive Control Center (PCC) is an easy-to-use, graphical tool designed to help you create and manipulate databases and control your DBMS. It allows you to access nearly all the functions of the product from one place. For a brief tour of PCC, see Chapter 3, "Using the Pervasive Control Center."

Utilities

A variety of graphical and command-line tools provide support for testing, configuring, and manipulating the many features and options provided by Pervasive.SQL. These tools are covered in-depth in *Advanced Operations Guide*, but a brief introduction to them is provided in "Helpful Utilities" on page 1-15.

Documentation

Pervasive.SQL comes with a printed copy of *Getting Started with Pervasive.SQL*, and with a complete set of online documentation. For more information about the documentation, see "Using Pervasive.SQL Documentation" on page 1-17.

Key Concepts

This section explains some basic concepts of databases and some of the key concepts that distinguish Pervasive.SQL from other database products.

Basic Database Structures and Terms

Most database management systems in use today share a common set of basic structures. This section briefly explains those structures. The descriptions that follow refer to the diagram below:

"Phone Book" Table

	Column 1	Column 2	Column 3	Column 4
Col Names	**Name**	**Address**	**Zip**	**Phone**
Row 1	Fred Black	643 Oak	12346	555-2345
Row 2	Jane Doe	112 Elm	12345	555-1212
Row 3	John Doe	112 Elm	12345	555-1212

Value

The most basic element of a database is a *value*. A value is one piece of data, one characteristic, for a specific entity. For example, in the diagram, the name "John Doe" or the phone number "555-1212" is a value.

Column or Field

Another element is a *column*, or a *field*. A column represents a characteristic with no specific value. Columns generally have names that describe the given characteristic. For example, in the telephone book, **Name** and **Phone** are columns. They do not have specific values unless you look up a particular person. Field is sometimes used to refer to the generic characteristic of a specific row. For example, someone might point to a specific box in the table above and ask, "What is the value of that field?"

Row or Record

Another element is called a *row*, or a *record*. A row is a collection of all the values for one particular instance. For example, one entry in

the phone book, complete with name, address, and phone number, is one record or row.

Cell

A *cell* is a column within a specific record. You can think of it as the intersection of a row and a column. Each cell has a specific value. For example, you might tell a co-worker, "The value of the cell located at row 2, column 3 is '12345'."

Table

A collection of rows and columns makes up a *table*. A table is a set of data that shares exactly the same structure. Tables generally have names that describe the contents of the table. For example, the table above is called "Phone Book." With Pervasive.SQL, each table is stored as a separate data file on the hard disk.

Index

An *index* is an ordered list of all the values in a particular column. A table can have zero or more indexes on it. The database engine uses indexes to find specific records in the database without having to step through every record one at a time. Creating indexes on columns which will frequently be used in database searches is likely to improve the performance of your database.

Database

A *database* is a collection of one or more tables. The data in the tables does not need to be related among the various tables, but usually there are many relations. For example, a database might consist of the "Food Preferences" table below, and the "Phone Book" table above. With Pervasive.SQL, a database consists of one or more data files and Data Dictionary Files (DDFs) on your hard disk. The DDFs are special data files that contain all the definitions for tables, columns, and other attributes that define the structure of your database.

Schema

The term *schema* refers to the complete set of definitions that describe the entire structure of a database. A typical schema includes

definitions for tables, columns, indexes, and many other attributes. The DDFs for a database contain the database's schema.

"Food Preferences" Table

	Column 1	Column 2	Column 3	Column 4
Col Names	**Name**	**Meat**	**Grain**	**Drink**
Row 1	Fred Black	sushi	wheat	sake
Row 2	Jane Doe	steak	oats	beer
Row 3	Ann Dean	cod	bran	spring water

Remote

The term *remote* refers to an object, such as a file server or a database, that is not located in the computer you are using now. When you connect to a database over the network, you are connecting to a remote database. Remote is the opposite of *local*. Remote can refer to either the client or the server, depending on whether you are currently seated at the server computer or a client computer. Remote always refers to an object that is not located on the system you are using.

Local

The term *local* refers to the computer you are using right now, or something stored on this computer. A local database is a database in which the data files are stored on the hard disk of the computer you are currently using. Local is the opposite of remote. Local can refer to either the client or the server, depending on whether you are currently seated at the server computer or a client computer.

Relational

The term *relational* refers to the storage of data in the form of related tables. The related tables allow relationships to be created between sub-sets of data.

For example, you can see that both our example tables contain the **Name** column, and some of the names are the same. Because we can cross-reference the names in the **Phone** table with the names in the **Food** table, we have the power to ask and answer such questions as, "What is the phone number of someone who likes steak?" We may

also answer such questions as, "Which consumer profile purchased the most product B after buying product A?"

You can see how powerful relational data access is. The SRDE component of Pervasive.SQL provides full relational access to your data.

Join

A *join* refers to an association between columns of related tables. Typically, a join operation is part of a SELECT query, which is used to obtain information from related tables.

Unique Benefits of Pervasive.SQL

One unique feature of Pervasive.SQL is that it allows applications to access data through either the industry-standard relational method outlined above, or through an ultra-high-speed transactional or hierarchical method known as the Btrieve interface. In fact, Pervasive.SQL allows applications to use both access methods at the same time to access the same data.

Transactional Interface

The transactional interface is a high-performance, low-overhead access method, capable of handling updates, inserts, and deletes much faster than other database products.

Applications that use the transactional interface bypass the relational interface and communicate directly with the MKDE to maximize performance.

In the interest of performance, the transactional interface offers only basic security, including file passwords and encryption. It does not allow SRDE data access to bypass transactional security.

Relational Interface

The relational interface uses industry-standard ODBC to provide a rich environment for data definition, security, reporting, stored procedures, and universal application access without requiring any application programming. Databases that are ODBC-enabled can be accessed by any ODBC-standard software program.

As an end user of an application based on Pervasive.SQL, you may not be able to choose which access method your application uses, but your application vendor has most certainly taken this into account.

No other DBMS available today offers this combination of flexible relational access and high-speed transaction throughput.

Additional Access Methods

In addition to the Transactional and Relational interfaces, Pervasive.SQL provides methods to access data through OLE DB, Java, ActiveX, ADO, and Pervasive Direct Access Components (PDAC) for Delphi and C++ Builder.

Terminology Revisited

When using the Btrieve interface, the terms *table* and *database* are generally not used, and data files are referred to directly as such. In addition, Btrieve users normally use the terms *records* and *fields* rather than *rows* and *columns*.

Why Pervasive.SQL?

You or your application vendor have chosen wisely in selecting Pervasive.SQL as the database of choice. Within the realm of small- to mid-sized enterprises, Pervasive.SQL provides the lowest-maintenance, highest-performance, best overall DBMS value on the market today.

Pervasive.SQL provides a number of advantages over other products. Here are just a few:

- *Lowest total-cost-of-ownership.* An independent study conducted by Aberdeen Group concluded that no major database product can match Pervasive.SQL's low total-cost-of-ownership.
- *No Database Administrator (DBA) required.* You can look in the newspaper any day of the week and see classified ads for Oracle, Sybase, or SQL Server database administrators, with sky-high salaries. Pervasive.SQL offers the unique Zero Database Administrator, or Z-DBA™, architecture. Its easy-to-use tools, bulletproof installation, and set-it-and-forget-it simplicity make Pervasive.SQL the perfect workhorse for desktop, workgroup, and departmental applications.
- *Scalable from the desktop to the Web.* Pervasive.SQL is available in two editions: the Ultra-light™ Workgroup database engine supports applications running locally on the same computer as the engine. It comes with a 5-user license and scales to 10 users. The Server engine starts with a 10-user license and scales to thousands of concurrent users, including intranet and extranet applications. Upgrading to another configuration requires no changes to the supported application, just plug-and-play with the new database engine.
- *Cross-platform support.* Unlike some other products, Pervasive.SQL does not lock you in to a single platform. Pervasive.SQL databases are binary-compatible and supported across Microsoft Windows, Novell NetWare, and several varieties of Linux. No matter where your data is or where it is going to be, Pervasive.SQL is there for you.
- *Big database features at a small price.* Pervasive.SQL offers full security, encryption, management and monitoring tools, and a host of other features you would expect to see in more expensive DBMS products.

- *Legendary stability and reliability.* Pervasive database engines have been serving the needs of business users for nearly twenty years. In fact, 70% of the Windows desktop accounting market uses Pervasive.SQL as the underlying DBMS of choice. When you must manage important data, you use the DBMS that will support your needs.
- *Multiple access methods.* Your application vendor can use the transactional interface for blazing performance on bulk data operations, while offering the richness of ODBC, OLE-DB, pure Java, and JDBC interfaces for data reporting, security, analysis, and standard compatibility. No other database management system offers all these access methods.

Understanding the Product Family

Pervasive.SQL is available in two different packages. The major differences between the packages are price and multi-user features.

- The Workgroup engine is least expensive, but it provides support only for small workgroup environments.
- The Server engine is designed for maximum scalability in high-volume, mission critical database applications where there is a dedicated database server. The Server engine quickly becomes most economical as you increase the number of users.

Both engines are fully compatible with any Pervasive.SQL database. To upgrade or downgrade from one package to another requires no changes to your application or to your database. Simply install the new package and you are ready to go. Both engines were designed with a common architecture and offer the same features.

In addition, there are several add-on packages that you can use to expand the capabilities of your database. This section explains the different packages.

Workgroup

The Workgroup engine offers a peer-to-peer network setup designed for stand-alone single-user installations up to small workgroups. The Workgroup engine is the only engine that offers multi-user access to Pervasive.SQL data located on a computer where no database engine is installed. For example, you can share data files stored on a NetWare or Windows server where no database server is installed, using a Workgroup engine installed on a different computer.

A major difference between Workgroup and Server is the Gateway feature of Workgroup. When there is no database engine running on the computer where the data is located, normally the first database engine to connect to that data handles all requests from other engines to access that data. This feature can be configured so that the same Workgroup engine always services that data, or the Gateway designation can be allowed to "float" based on which Workgroup engine connects to the data first during any given work day.

The Workgroup engine comes with a 5-user license. Additional users may be added.

Server

The Server engine offers a full client/server architecture providing excellent performance and scalability for up to thousands of concurrent users. The Server engine can be monitored and configured remotely.

The Server engine must be located on the same computer as the data files it is intended to access.

The table below compares Server and Workgroup features.

Table 1-1 Comparison of Server and Workgroup Features

Feature	Server	Workgroup
Supports Btrieve, ODBC, OLE-DB, Java, JDBC, and ActiveX interfaces	✓	✓
Full-featured relational support (online backup, security, referential integrity, management tools)	✓	✓
Binary compatible data files across all platforms and engine editions	✓	✓
Easy, compatible upgrading, no application changes required	✓	✓
Includes complete online documentation	✓	✓
Can access data on a file server where no database engine is installed		✓
Supports remote ODBC client connections	✓	✓
Requires a Workgroup engine on all computers expected to access remote data	N/A	✓
Engine runs on Windows	✓	✓
Engine runs on NetWare	✓	
Engine runs on Linux	✓	
Multi-user for small groups	✓	✓
Scales to thousands of users	✓	
Extranet license available	✓	
Security	✓	

DataExchange

Pervasive.DataExchange is a separate product that allows you to synchronize two or more databases that have the same structure and are always or occasionally connected by a network.

This type of product is often used to keep the data of traveling employees synchronized with the home database, or to allow regional offices to maintain fully synchronized copies of the corporate database. The synchronized copies do not require having to download the entire database to each location.

Pervasive.DataExchange offers solutions for data continuity, data synchronization, and data portals. For more information about DataExchange, contact your sales representative or visit the Pervasive Web site at http://www.pervasive.com.

Crystal Reports for Pervasive.SQL

Another optional product, Crystal Reports provides rich capabilities for creating and formatting reports based on Pervasive.SQL databases. Reports can be customized in thousands of ways and published as HTML, Microsoft Word document, Microsoft Excel document, or other formats.

For more information about Crystal Reports, contact your sales representative or visit the Pervasive web site at: http://www.pervasive.com.

Helpful Utilities

Pervasive.SQL comes with a variety of utilities designed to help you control and manage your databases. Most of the utilities run only on 32-bit Windows and allow remote function to NetWare or Linux database server engines.

Table 1-2 Summary of Pervasive.SQL Utilities

Utility name	Runs on these platforms	Description	Server or Workgroup
Pervasive Control Center	Win32	Utilities warehouse for Pervasive.SQL. Lists engines and databases available.	All
Configuration	Win32	Manipulates settings for Pervasive client and server components.	All
Monitor	Win32	Monitors server engine activity. Useful for database administration and programming diagnostics.	All
Function Executor	Win16, Win32	Executes Btrieve operations, enabling you to learn how the Btrieve interface operates on data files.	All
Pervasive.SQL Maintenance	Linux, Win32, and NetWare	Performs common Pervasive.SQL file and data manipulations, such as importing and exporting data.	Linux, Windows, and NetWare servers All 32-bit Windows workstations for GUI version
SQL Data Manager	Win32	Allows you to execute SQL statements interactively.	All (local) Server (remote)
Table Designer	Win32	Allows you to add, delete, or change the characteristics of columns within a table, and to create a database or a table.	All (local) Server (remote)

Table 1-2 Summary of Pervasive.SQL Utilities

Utility name	Runs on these platforms	Description	Server or Workgroup
Rebuild	Win16, Win32, and NetWare	Converts one version of MicroKernel files into another version.	Windows and NetWare servers Windows 32-bit workstations
License Administrator	Win32, NetWare, and Linux	Manages Pervasive.SQL license keys and user counts.	Windows, NetWare, and Linux servers (remote) Workgroup (local) Command-line version on Windows, NetWare, and Linux
ODBC Administrator	Win32	Sets up Data Source Names (DSNs) for client and engine interfaces.	All Windows engines (local) Server (remote)
Gateway Locator	Win32	Used to configure and maintain gateway locator files for the Workgroup engine.	Workgroup engine only
Pervasive System Analyzer	Win32	Analyzes system components, runs communication tests, and archives or restores previous database engine files on your system.	All

Using Pervasive.SQL Documentation

All Pervasive.SQL documentation, both printed and online, assumes you are familiar with the basics of using a computer, such as clicking and dragging, opening and saving files. If you need assistance with these tasks, please consult the documentation that came with your computer and/or operating system.

This section describes the Pervasive.SQL documentation. Of these titles, only *Getting Started* and *Status Codes Quick Reference* card are provided in hardcopy with Pervasive.SQL; the rest are provided on the Pervasive.SQL CD-ROM. These online documentation files are installed on Windows platforms when you choose the **Typical** installation procedure. They are available as an option in the **Custom** installation procedure. The content is accessible through the **Start** menu:
Programs ▸ Pervasive.SQL V8 ▸ Documentation ▸ Pervasive.SQL V8.

For NetWare users, you can access the online documentation on any Windows workstation that has the database client installed.

For Linux users, you can access the online documentation on the server in HTML format. Certain utilities also provide manpages.

You can order printed copies of the documentation from your sales representative.

Getting Started Guides

Getting Started with Pervasive.SQL (Server Edition) and *Getting Started with Pervasive.SQL (Workgroup Edition)* help you get Pervasive.SQL running with installation, setup, and troubleshooting information. Both *Getting Started* editions cover the following topics:

- Preparing to install Pervasive.SQL V8
- Installing Pervasive.SQL V8
- Upgrading from previous versions of Pervasive.SQL or Btrieve
- Configuring Pervasive.SQL V8
- Troubleshooting your Pervasive.SQL V8 installation
- Where to go for Pervasive.SQL product information and technical support

What's New

Pervasive.SQL What's New provides an overview of the new features and changed behavior for the current release of Pervasive.SQL relative to the most recent previous release. This book provides summary information on new features and directs you to the locations in the documentation where the new features are fully discussed.

User's Guide

Pervasive.SQL User's Guide introduces Pervasive.SQL and describes common user tasks. The guide discusses the database engine, Pervasive.SQL utilities and other key components; the differences between Server and Workgroup engines; and the differences between ODBC and Btrieve access. *User's Guide* provides you with the basics to work with Pervasive.SQL successfully.

Advanced Operations Guide

Advanced Operations Guide provides detailed information at the administrative level, including the steps to perform common procedures and several new ones. Topics include:

- checking database consistency
- performing periodic backups
- configuring network protocols and understanding network topologies
- working with database security
- basic configuration guidelines
- configuration options reference
- moving, renaming, compacting and rebuilding files

SQL Engine Reference

SQL Engine Reference gives database programmers a complete reference guide to the SQL relational database language. It also covers SQL engine parameters and limitations.

Status Codes and Messages

Status Codes and Messages documents all possible status codes and numbered messages that can be received when using Pervasive software.

The *Status Codes and Messages Quick Reference* card is also included with your complete documentation set.

Pervasive Products and Services

Pervasive Products and Services provides an outline of how to work with Pervasive Software and describes the Pervasive.SQL database family.

Online Help

Pervasive.SQL comes with a full set of online documentation, available in the Pervasive program group on your **Start** menu.

You can download other formats and related documentation from this address:

http://www.pervasive.com/support/technical/online_manuals.asp

You can also find additional information on the Web site at:

http://www.pervasive.com/support/DataSheets.asp

http://www.pervasive.com/library

File System Security

The Pervasive.SQL engine adheres to the file system security defined by the specific operating system (OS), such as Windows NT File Sharing or Novell Storage Services.

Only the Server engine can enforce OS-level file security based on the privileges assigned to the login user name. The Workgroup product does not attempt to do this. In a small office, where Workgroup engines are most common, this can be considered a plus because they are usually short on networking experts, and the fewer barriers to successful data access the better.

chapter

2

Using Pervasive.SQL

A Walk-through of Basic User Tasks

If you have not already done so, install Pervasive.SQL V8 by following the instructions in *Getting Started with Pervasive.SQL*.

This chapter cover the basic tasks you need to know to work with Pervasive.SQL databases.

This chapter includes the following sections:

- "Starting and Stopping the Database Engine" on page 2-2
- "Granting Administrative Rights for the Database Engine" on page 2-10
- "Setting Up ODBC Database Access" on page 2-21
- "Setting Up Database Access on Windows" on page 2-25
- "Setting Up Database Access on a NetWare Server" on page 2-31
- "Setting Up Database Access on a Linux Server" on page 2-37
- "Setting Up Client Access" on page 2-40
- "Accessing Data via ODBC From Other Applications" on page 2-46
- "Deleting DSNs" on page 2-52
- "Bound Databases and Enforced Integrity" on page 2-55
- "Using the Fast User Switching Feature of Windows XP" on page 2-56

Starting and Stopping the Database Engine

This section outlines how to start and stop the Pervasive.SQL engine. For most engine configuration parameters, you need to stop and restart the engine in order for a particular change in your configuration to take effect.

To start and stop the database engine, follow the instructions for your platform:

- "Starting and Stopping the Server Engine on a Windows Server" on page 2-3
- "Starting and Stopping the Database Engine on NetWare" on page 2-8
- "Starting and Stopping the Database Engine on Linux" on page 2-9

Note ***Btrieve v6.15 Users:*** If you have any log (.LOG) files you want to roll forward, you must do so before you load the Pervasive.SQL V8 MicroKernel, which uses a different logging scheme. You must use both the v6.15 or earlier engine and its accompanying Roll Forward Utility. Refer to your Btrieve v6.15 or earlier documentation for information about logging and instructions about how to roll files forward.

Starting and Stopping the Server Engine on a Windows Server

On Windows NT or 2000, Pervasive.SQL Server runs as services. The services are loaded as part of the installation process and are set to be always available if you followed the Typical installation.

➤ To start the database services on Windows NT or 2000 by using services

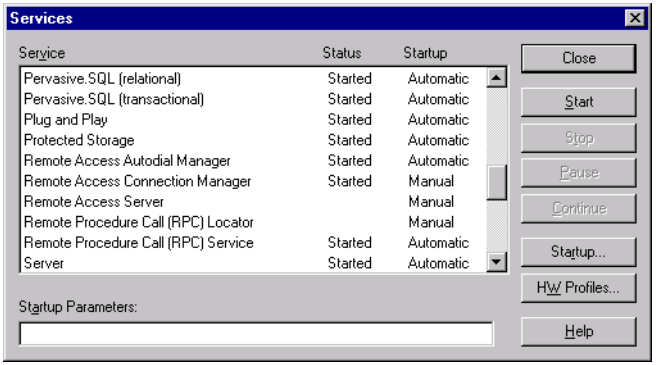
1 Click **Start** ▸ **Settings** ▸ **Control Panel**.

2 Do one of the following:

a. On Windows NT, double-click on the **Services** icon.

A dialog box similar to Figure 2-1 displays:

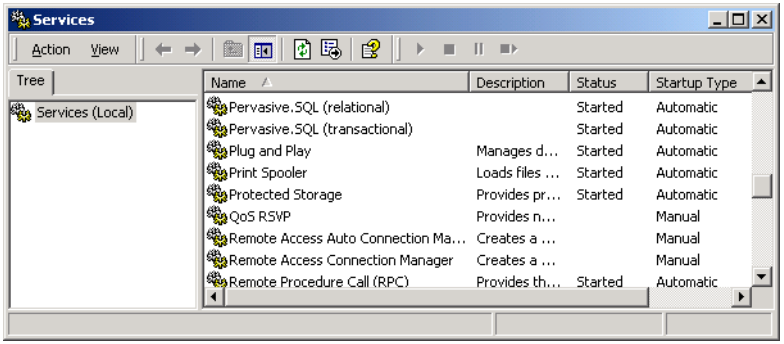
Figure 2-1 Windows NT Services Dialog Box



b. On Windows 2000, double-click **Administrative Tools** then double-click **Services**.

A dialog box similar to Figure 2-2 appears.

Figure 2-2 Windows 2000 Services Dialog Box



3 Do one of the following:

   a. On Windows NT, click **Pervasive.SQL (relational)** then click **Start**. Click **Pervasive.SQL (transactional)** then click **Start**.

   b. On Windows 2000, right-click **Pervasive.SQL (relational)** then click **Start**. Right-click **Pervasive.SQL (transactional)** then click **Start.**

➢ To stop the database services on Windows NT or 2000 by using services

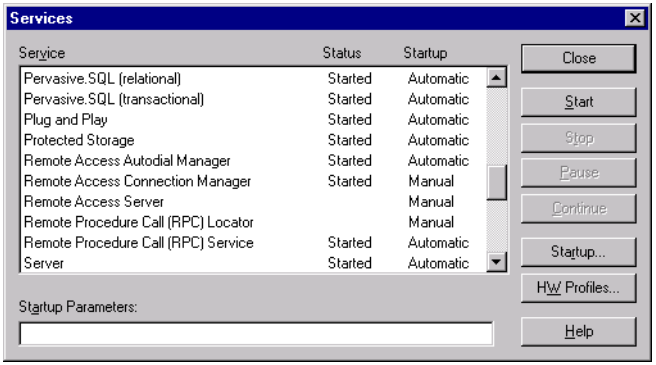
1 Click **Start** ▸ **Settings** ▸ **Control Panel**.

2 Do one of the following:

   a. On Windows NT, double-click on the **Services** icon.

   A dialog box similar to Figure 2-1 displays:

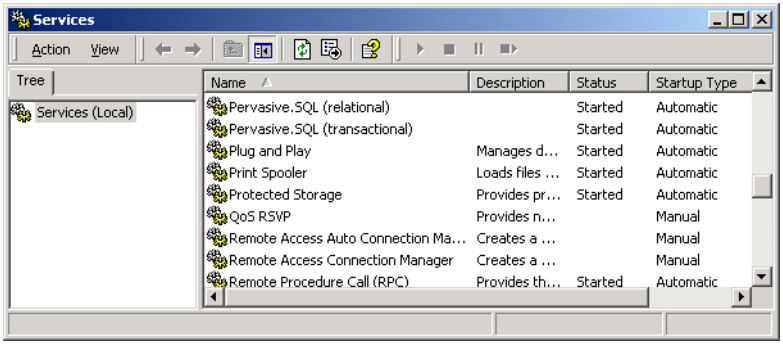
Figure 2-3 Windows NT Services Dialog Box



b. On Windows 2000, double-click **Administrative Tools** then double-click **Services.**

A dialog box similar to Figure 2-2 appears.

Figure 2-4 Windows 2000 Services Dialog Box



3 Do one of the following:

a. On Windows NT, click **Pervasive.SQL (relational)** then click **Stop**. Click **Pervasive.SQL (transactional)** then click **Stop**.

b. On Windows 2000, right-click **Pervasive.SQL (relational)** then click **Stop**. Right-click **Pervasive.SQL (transactional)** then click **Stop.**

➤ To start the database services on Windows NT or 2000 by using Net Start or Btisc.exe

1. Click **Start** ▸ **Run**.
2. Type one of the following:
   a. `net start "Pervasive.SQL (<transactional | relational>)"`
   b. `btisc start "Pervasive.SQL (<transactional | relational>)"`
3. Click **OK**.

Note Btisc.exe is located in the \PVSW\BIN folder.

➤ To stop the database services on Windows NT or 2000 by using Net Start or Btisc.exe

1. Click **Start** > **Run**.
2. Type one of the following:
   a. `net stop "Pervasive.SQL (<transactional | relational>)"`
   b. `btisc stop "Pervasive.SQL (<transactional | relational>)"`
3. Click **OK**.

Note Btisc.exe is located in the \PVSW\BIN folder.

➤ To start the database services on Windows NT or 2000 by using PCC

1. Click **Start** ▸ **Programs** ▸ **Pervasive.SQL V8** ▸ **Control Center**.
2. In the Namespace, click on a registered engine.
3. Click **Action** ▸ **Tasks**.
4. Do one of the following:
   a. Click **Start All Pervasive Services** to start both the relational and transactional services.
   b. Click **Start Relational Service** to start the relational service.

c. Click **Start Transactional Service** to start the transactional service.

➢ To stop the database services on Windows NT or 2000 by using PCC

1 Click **Start** ▸ **Programs** ▸ **Pervasive.SQL V8** ▸ **Control Center**.

2 In the Namespace, click on a registered engine.

3 Click **Action** ▸ **Tasks**.

4 Do one of the following:

a. Click **Stop All Pervasive Services** to stop both the relational and transactional services.

b. Click **Stop Relational Service** to stop the relational service.

c. Click **Stop Transactional Service** to stop the transactional service.

Starting and Stopping the Workgroup Engine on Windows

➢ To start the Workgroup Engine on Windows

1 Click **Start** ▸ **Programs** ▸ **Pervasive.SQL V8** ▸ **Engines** ▸ **Pervasive.SQL Workgroup Engine**.

➢ To stop the Workgroup Engine on Windows

1 On the Windows taskbar, right-click the **Pervasive Database** icon: .

2 Click **Stop Engines and Exit**.

Note You will receive a warning message when trying to stop the engine if any of the following is true:

- There are active clients.
- No activity took place since the engine loaded.
- 10 seconds has not elapsed since the last operation took place.

Starting and Stopping the Database Engine on NetWare

➤ To start the Database Engine on NetWare

Note You must load AFTER311.NLM before you load the MicroKernel.

Enter each of the following commands separately from the console window:

```
BSTART
MGRSTART
```

Pervasive.SQL supports NetWare Storage Services (NSS) volumes on NetWare 5.0 and up. NSS volumes must be mounted before starting the database engine. That is, before accessing the server with the database engine.

As an example:

```
LOAD NSS
MOUNT ALL
SYS:ETC\INITSYS.NCF
BSTART
MGRSTART
```

Also, database updates performed against data files on NSS volumes may run more slowly than with earlier versions of NetWare. As noted in Novell TID 2952147 (http://www.novell.com), "NSS is optimized for reading files." Updates "will almost always perform a little faster on the legacy file system."

Based on this information, you may wish to store frequently-updated data files on regular NetWare volumes rather than NSS volumes.

➤ To stop the Database Engine on NetWare

Enter each of the following commands separately from the console window:

```
MGRSTOP
BSTOP
```

Note Never enter the **BSTOP** command before the **MGRSTOP** command.

Tip If NetWare does not allow you to stop the engines with these commands because of dependencies on other modules, first enter the command BTRV UNLINK, then perform the commands noted above.

Starting and Stopping the Database Engine on Linux

In Linux, the database engine runs as a daemon. The daemon is loaded as part of the installation process and is set to be always available if you followed the Typical installation.

You must be logged in as the `root` user to start and stop the Pervasive.SQL V8 daemon process. We recommend that you use the shell script `psql` to start and stop the process. However, you may start and stop individual daemons by using `mkded` and `sqlmgr`.

➤ To start the Database Engine on Linux

Enter the following at the command line:

```
/etc/rc.d/init.d/psql start
```

➤ To stop the Database Engine on Linux

Enter the following at the command line:

```
/etc/rc.d/init.d/psql stop
```

Granting Administrative Rights for the Database Engine

This section begins by outlining those Pervasive.SQL V8 tasks that require administrative-level access at the operating system level and those that do not. The section then walks you through the steps to grant a user administrative-level access for each of the supported operating systems.

Note This section only applies to the Server engine unless otherwise noted.

Tasks Requiring Administrative Rights

Administrative-level rights are required to:

- create and configure named databases and tables
- view or modify a table design with Table Designer
- set engine configuration options
- view and set engine monitoring values
- view certain engine configuration settings

How Administrative Rights are Granted

To have administrator-level access you must:

- possess full administrator-level rights *on the machine on which the database engine is running* (a domain administrator, for example, may lack full permissions on certain local machines), or
- be a member of the operating system group Pervasive_Admin.

To modify a table design with Table Designer, you must have full administrator rights on the machine on which the database engine is running *even if* you are a member of the Pervasive_Admin group.

Note For Linux servers, administrator-level rights can be granted only by using the btadmin utility to add users and passwords to the btpasswd file.

The Pervasive_Admin option is offered so that you can grant users administrative rights to the database engine without granting them administrative rights to the operating system where the database engine resides.

Rights Within an Active Directory Environment

You may use one or more Pervasive_Admin groups within an Active Directory environment. See "Support for Active Directory Service" on page 7-2 in *Getting Started with Pervasive.SQL* (*Server* edition) or "Support for Active Directory Service" on page 8-2 in *Getting Started with Pervasive.SQL* (*Workgroup* edition).

Rights Provided to non-Administrative Users

Runtime-only access enables a user without administrator-level rights to perform such functions as:

- extract a list of DSNs
- extract a count of DSNs
- extract information on a DSN
- extract information on the location of the DBnames configuration file (dbnames.cfg)
- connect to databases
- retrieve, update, insert, and delete data (as permitted by database security)

Tasks for Granting Administrative Rights

To grant a user administrative rights, follow the instructions for your platform:

- "Granting Administrative Rights on a Windows Server" on page 2-12
- "Granting Administrator Rights on NetWare 4.2, 5.x or 6.x" on page 2-19
- "Granting Administrator Rights on Linux" on page 2-19
- "Logging in as Administrator on any platform" on page 2-20

Granting Administrative Rights on a Windows Server

Users who are members of **Pervasive_Admin** or of **Administrators** are permitted to perform administrative tasks on the database engine.

➢ To grant a user database administrator rights on Windows NT

Note You must be logged onto the Windows server as a user with full administrator-level rights on the server or be a member of the Pervasive_Admin group defined on the server.

1. Open the Windows NT User Manager (click **Start ▸ Programs ▸ Administrative Tools ▸ User Manager.**)

   The graphical user interface for the User Manager appears.

2. Click **User ▸ New Local Group**. A dialog box appears.

3. Type in `Pervasive_Admin` as the group name.

   You may now add users to this list. (Click **Add**, select user, click **Add**, click **OK**.)

4. Click **OK** when finished.

Note If the **Log on as** setting for the Pervasive.SQL services is not **System Account**, see "Services Settings and Log In Authority" on page 2-13.

➢ To grant a user database administrator rights on Windows 2000

Note You must be logged onto the Windows server as a user with full administrator-level rights on the server or be a member of the Pervasive_Admin group defined on the server.

1. Click **Start ▸ Settings ▸ Control Panel ▸ Users and Passwords**.

2. Click the Advanced tab. In the area "Advanced User Management," click **Advanced**.

3. Click the **Groups** folder. From the menu, click **Action ▸ New Group**.

4 Type in Pervasive_Admin as the group name.

(To add users to this group, click Add, select user name, click Add > OK.)

5 Click **Create** to create the group.

6 Click **Close**.

Note If the **Log on as** setting for the Pervasive.SQL services is not **System Account**, see "Services Settings and Log In Authority" on page 2-13.

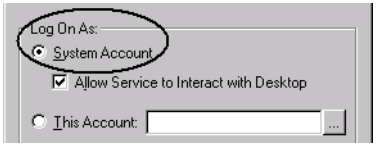
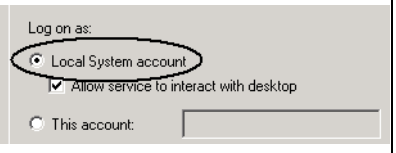
Services Settings and Log In Authority

Certain operating system settings for the Pervasive.SQL services must be in effect for you to log in to the machine running the database engine. These settings apply whether or not you use a Pervasive_Admin user group.

The settings apply to the Pervasive.SQL Server engine and to the Workgroup engine if you are running the Workgroup engine as a service. See "Running the Workgroup Engine as a Service" on page 10-15 in *Getting Started with Pervasive.SQL* (*Workgroup* edition).

Default Setting

By default installation, both the Transactional and Relational services set **Log on as** to **System Account** (**Local System Account** on Windows 2000).

Windows NT	Windows 2000
	

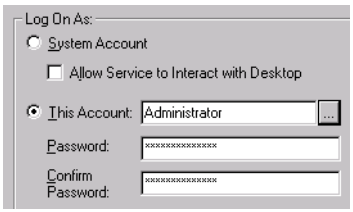
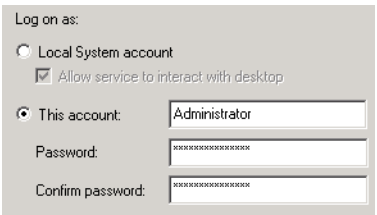
Logging On as "This Account"

If you change the **Log on as** setting to **This account**, you must change the user rights policy **Act as part of the operating system** for the account. Otherwise, remote login fails.

For example, both the Monitoring utility and the Configuration utility require that you log in to the operating system on the machine where the database engine is running. You will receive a message that login failed if the account specified for **This account** cannot act as part of the operating system.

Note that even the Administrator account requires that you set the user rights policy for **Act as part of the operating system**.

You specify **This account** on the services property sheet.

Windows NT	Windows 2000
	

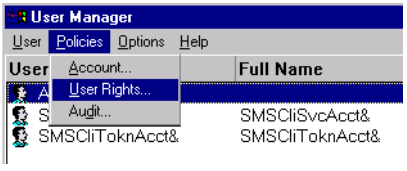
User Rights Policy Tasks

The following tasks explain how to change the user rights policy.

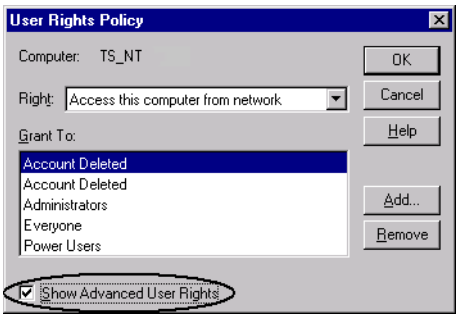
➢ To Set User Rights Policy on Windows NT

1 Click **Start** ▸ **Programs** ▸ **Administrative Tools** ▸ **User Manager**.

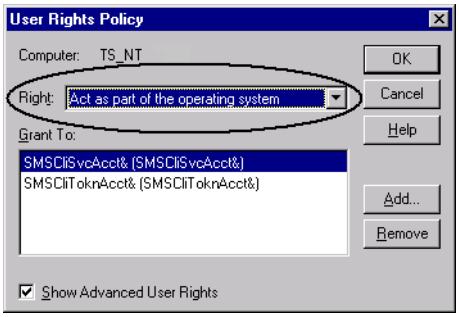
2 Click **Policies** ▸ **User Rights**.



3 Click the option **Show Advanced User Rights**.



4 In the **Right** selection list, click **Act as part of the operating system**.



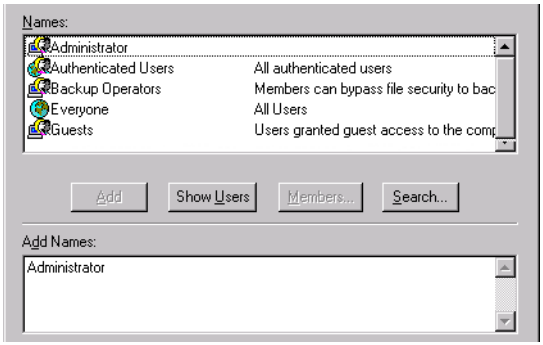
5 Click **Add. . .**

The Add Users and Groups dialog appears.

6 In the **Names** list, click on the user or group for whose account you want to grant the user policy. (For example, you could grant the policy to the Pervasive_Admin group.)

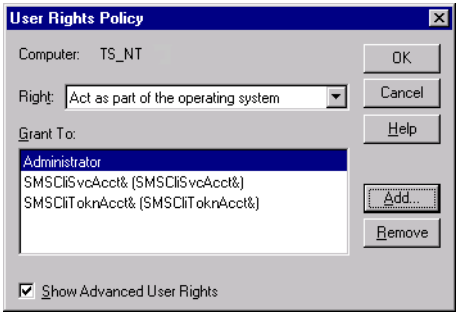
7 Click **Add**.

The user name is added to the **Add Names** list. For example, the following image shows that the Administrator has been added.



8 Click **OK**.

The user account is added to the **Grant to** list.



9 Click **OK**.

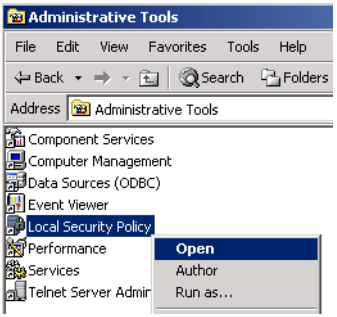
10 Exit User Manager.

➢ To Set User Rights Policy on Windows 2000

1 Click **Start** ▸ **Settings** ▸ **Control Panel**.

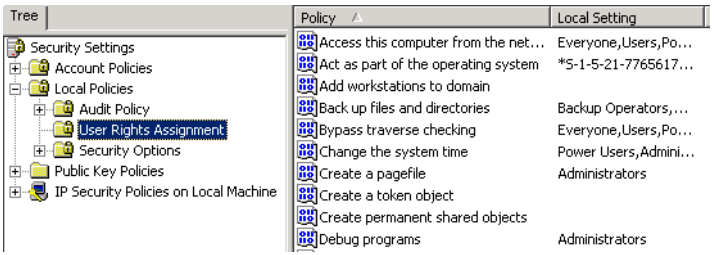
2 Double-click **Administrative Tools**.

3 Right-click **Local Security Policy**, then click **Open**.

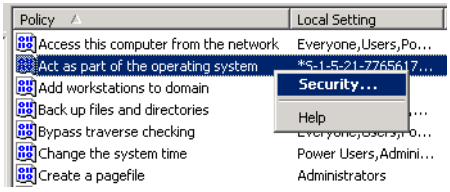


4 Expand the tree for **Local Policies** (click the plus sign).

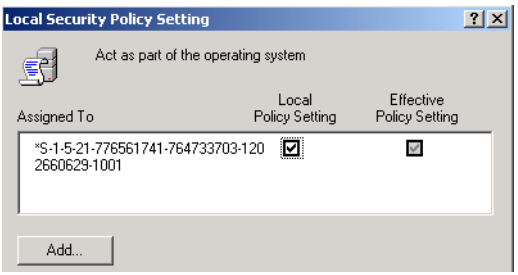
5 Click on **User Rights Assignment**.



6 In the policy pane, right-click **Act as part of the operating system**, then click **Security**.



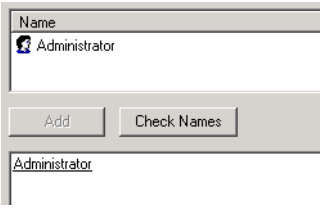
7 Click **Add**.



8 In the **Name** pane, click on the user or group for whose account you want to grant the user policy. (For example, you could grant the policy to the Pervasive_Admin group.)

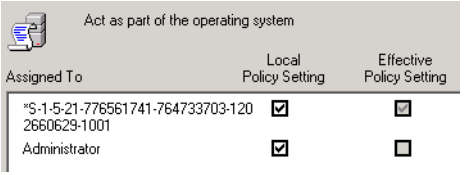
9 Click **Add**.

The user name is added to the bottom pane. For example, the following image shows that the Administrator has been added.



10 Click **OK**.

The user name is added to the settings for local security policy.



11 Click **OK**.

12 Exit the window for **Local Security Settings**, then exit **Administrative Tools**.

Granting Administrator Rights on NetWare 4.2, 5.x or 6.x

➤ To grant a user database administrator rights on NetWare 4.2, 5.x or 6.x

1. Do one of the following:
   a. Run the 32-bit utility **NWADMN32** located in SYS:PUBLIC\WIN32.
   b. Run the **NWADMN3X** utility located in SYS:PUBLIC.

   The NetWare Administrator dialog box appears.
2. Select the **Object** menu.
3. Select **Create.** The **New Object** dialog box opens.
4. Select **Group** and click **OK**. The Create Group dialog box opens. Enter Pervasive_Admin as the name of the group, and click **Create**.
5. Once Pervasive_Admin is created, you can double-click on the name, and the **Details** dialog box for that group will appear. **Members** is an options page (listed in the right-hand panel of the dialog box); click on it and then click **Add** to select a user and click **OK** to close the dialog box.

Note Users may not log in to Pervasive.SQL V8 if they are in violation of any restrictions set in their NetWare user profile. These restrictions may be accessed and changed by double-clicking the user name in the NDS tree, which will bring up the **Details** dialog box for that user. The **Login Restrictions**, **Login Time Restrictions** and **Network Address Restrictions** options pages appear in the right-hand panel of the dialog box; these may be reviewed to ensure no violations are occurring.

Granting Administrator Rights on Linux

➤ To grant a user administrator rights on Linux

A user cannot remotely administer a server engine on Linux unless the user has first been set up as a database user with administrative rights. You can perform this task by using the btadmin utility at the server command line.

See also “Pervasive.SQL Account Management on Linux” on page 14-2 in *Getting Started with Pervasive.SQL* (Server edition) for a complete discussion of configuring the environment for administrative rights.

1 Login to the Linux server as `psql` (or as `root` if the `PATH` and `LD_LIBRARY_PATH` variables have been set and exported). No other user is permitted to run btadmin.

2 Create a new user with administrative rights by running btadmin:

```
btadmin -p passwd a+ user_name
```

For example, if you wanted to create an administrative user "tim" with password "tim56", you would enter the following command:

```
btadmin -p tim56 a+ tim
```

Note Users created with btadmin are not related to Linux system users. These users are known only to the database engine.

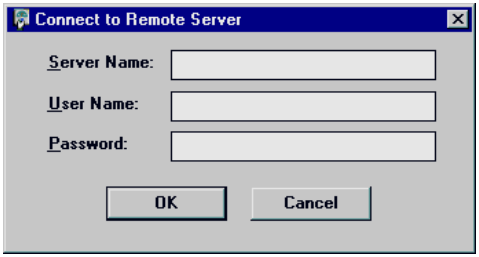
Logging in as Administrator on any platform

➢ To connect to a remote Pervasive.SQL V8 server

1 Use the Monitor or Configuration utility to connect to a remote server. Refer to the *Advanced Operations Guide* for a discussion of these utilities.

A dialog box appears. Enter your operating system user name and password, and click **OK**.

Figure 2-5 Connect to Remote Server Dialog Box



The password is encrypted before being sent over the network using a unique and pre-defined encryption key. The Pervasive.SQL V8 engine unpacks and decrypts the user name and password, and verifies access. It then returns a status code to the client indicating the success or failure of the verification.

Setting Up ODBC Database Access

This section covers the steps to set up access to a database. We will review some conceptual information before detailing the steps to perform the tasks of setting up Client and Engine (Server) DSNs and creating a new database using Pervasive.SQL V8.

This section covers the following topics:

- "Basic Concepts" on page 2-21
- "What to Know if You are Recreating DSNs" on page 2-23
- "Prerequisites to Setting up Database Access" on page 2-24

Basic Concepts ODBC Standard

Pervasive.SQL adheres to the Microsoft standard for ODBC database connections. According to the standard, applications must connect to databases through Data Source Names (DSNs) defined in the operating system.

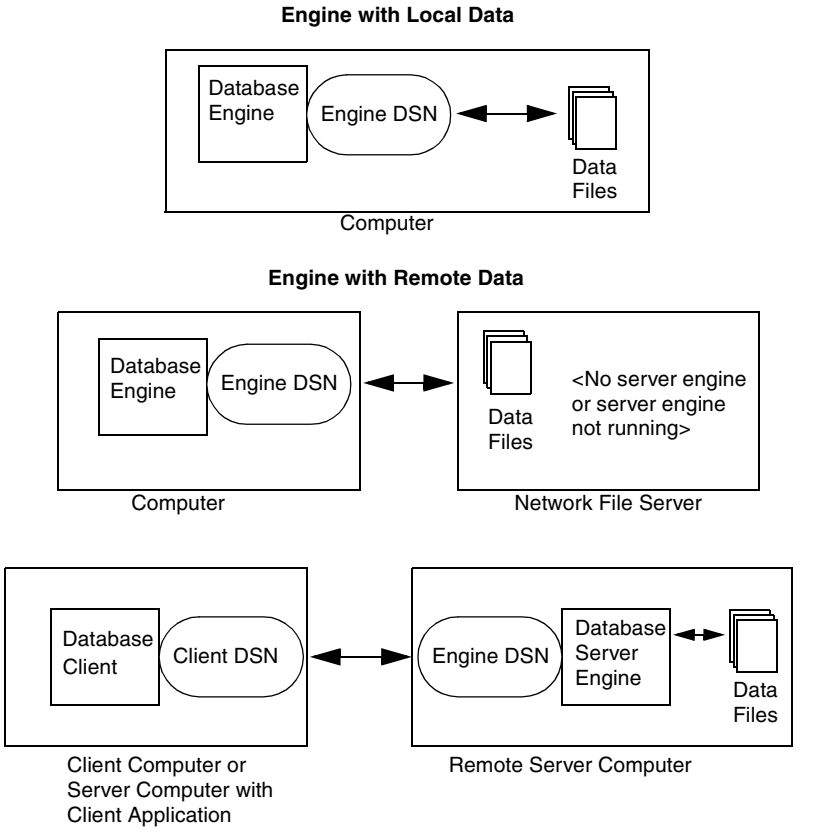
Note Pervasive.SQL does not support File DSNs. You must use User or System DSNs. System DSNs are generally preferred, because they are available to all users on a given computer.

Every Pervasive.SQL database that you expect to access using ODBC must have a DSN available on the same computer as the database engine, and (if applicable) another DSN on the client computer. The only exceptions are Pervasive tools, which can access remote databases without using DSNs on the client computer. A DSN that points to a database engine is called an *Engine* DSN. A DSN that points to an engine DSN is called a *Client* DSN.

Note Pervasive.SQL databases that are accessed only through Btrieve do not need DSNs. However, in this case, the database is not visible in PCC nor can it be manipulated using PCC. Pervasive recommends using Data Dictionary Files (DDFs) and DSNs with all databases, including ones accessed only through Btrieve, to make them easier to manipulate.

Figures 2-6 shows possible DSN configurations.

Figure 2-6 Example DSN Configurations



Servers and Clients

Pervasive.SQL servers are also clients. The client components of Pervasive.SQL are installed with every Server engine or Workgroup engine. So you can use your server machine to connect to other servers as a client. *Pervasive.SQL clients* can connect to remote machines where a Pervasive.SQL Server engine is installed.

Data Source Names

The client-server architecture calls for the naming of each specific data set so that it can be referred to by a well-known name. There are generally three ways to create DSNs:

1 Create an Engine DSN from the server console.

2 Create an Engine DSN remotely from a client machine.

Note On NetWare, the *only* way to create an Engine DSN is to create it remotely from the client.

3 Create a Client DSN on each client machine.
While Pervasive tools can access remote databases without a client DSN present on the client machine, ODBC-based applications such as Microsoft Excel and Microsoft Access cannot do so. You must create a client DSN on each client computer that needs to access network databases from local applications.

Components of a Database Name

There are two components of a database name. Pervasive.SQL uses an internal Database Name (DBNAME) to identify the location of the Data Dictionary Files (DDFs) and the data files for each database. An ODBC Data Source Name (DSN) entry refers to one DBNAME.

You may set up more than one DSN that refers to the same DBNAME. If the physical location of the data files on the server is changed, only the DBNAME needs to be updated. All DSNs remain unchanged.

What to Know if You are Recreating DSNs

Btrieve application users

Applications that only use Btrieve do not require you to follow the procedures in this section. However, you must follow these procedures to access Pervasive.SQL through ODBC. This includes the SQL Data Manager utility and all Microsoft applications. If you are not sure whether your application uses ODBC, check with your network administrator.

Pervasive.SQL v7 users

You must recreate all DSNs created with Pervasive.SQL 7 to access them in Pervasive.SQL V8. However, you do not need to rename existing database names. To re-create DSNs, follow the instructions provides in “Deleting DSNs” on page 2-52.

Prerequisites to Setting up Database Access

Before you begin, verify the following:

- The data files and DDFs already exist. If these do not exist, ask your network or systems administrator to create them, or refer to *Advanced Operations Guide* for information on creating a database.
- If the data files are located on a remote server, you must know the name of the server. You must also know the location of the data files on the server. If the database already has a DBNAME, this information can be used instead of the location of the data files. If you do not have this information, see your network or systems administrator.
- If the data files are located on a remote server and an Engine DSN has not been created on the server, you must possess administrator rights on the remote machine that houses the database you wish to access, to set up an Engine DSN. Setting up local access requires OS system rights to create a System DSN.

The sections that follow allow you to set up database access as long as the conditions above are met. The following topics are provided:

- "Setting Up Database Access on Windows" on page 2-25
- "Setting Up Database Access on a NetWare Server" on page 2-31
- "Setting Up Database Access on a Linux Server" on page 2-37
- "Setting Up Client Access" on page 2-40

Setting Up Database Access on Windows

➤ To Set Up Database Access on Windows

Note If you have not done so already, review "Prerequisites to Setting up Database Access" on page 2-24 before proceeding.

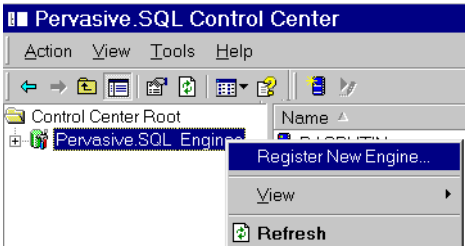
1 Open Pervasive Control Center (PCC):
Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center.

2 Double-click on **Pervasive.SQL Engines**.

The PCC Namespace contains a list of database engines that have been registered.

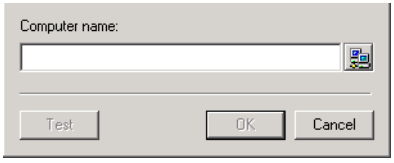
If you do not see the name of the database engine to which you want to connect, you must register the remote engine with PCC. To do so, right-click **Pervasive.SQL Engines** then click **Register New Engine**.

Figure 2-7 Registering a New Engine



A dialog is displayed that allows you to choose the machine name where the Pervasive.SQL engine resides.

Figure 2-8 Choosing a Computer Name

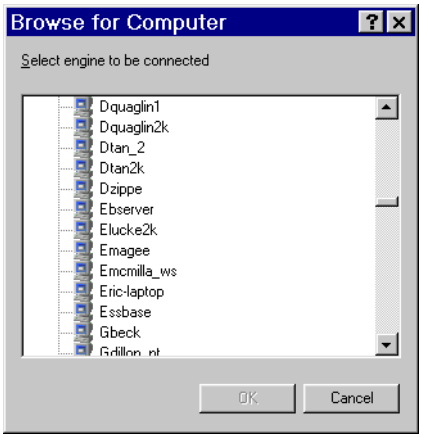


3 Do one of the following:

a. Type a computer name in the **Computer name** field and click **OK**.

b. If you are unsure of the computer name, click the button located to the right of the name field and browse from the Network list that appears. Click on the desired computer name, then click **OK**.

Figure 2-9 Choosing a Computer Name from a Network List



4 Back in the main screen of PCC, double-click the icon representing the engine you are working with. Then double-click the **Databases** folder associated with the selected engine.

5 Inspect the databases that are listed to determine whether the database you wish to access has already been set up with an Engine DSN.

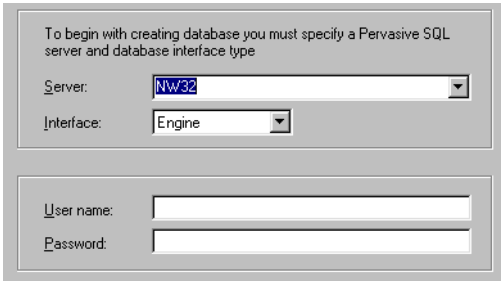
If the database you wish to access is visible, then it already has an Engine DSN. You are finished with this procedure. Go to "Setting Up Client Access" on page 2-40 for instructions on how to set up the client machines with Client DSNs.

If you do **not** see the database you wish to access, proceed with step 6.

6 Create a New Database for the engine:
Right-click on the database folder underneath the engine name, then click **New Database**.

If you are working with a Server engine and you are not logged into the server as a user with system administrator permissions, you see the login screen shown below.

Figure 2-10 Create an Engine DSN and Connect to a Server Database



If you are using Workgroup engine, or you are using the server console or are already logged into the server with system administrator permissions, skip to step 8 as shown in Figure 2-11.

7 The **Server:** field should indicate the server that you are currently working with. Choose Engine for the **Interface** type, and enter your operating system user name and password for that server, in the **User name** and **Password** boxes.

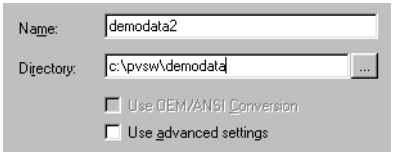
Remember that you must have full administrator-level rights on the server operating system to be able to complete this task.

Click **Next**.

8 In the **Name** field, specify an Engine DSN name for the database. This name will appear in the database listing in PCC after you complete this task.

In the **Directory** field, specify the location of the data files.

Figure 2-11 Naming the New Database on the Server



Note If you are working with a Server engine, you must specify a full path (not a mapped drive) that makes sense on the server. The Server engine cannot interpret mapped drive paths. For example, if you have a mapped drive S:\data1\datafiles on your client, referring to the folder C:\servers\data1\datafiles on the server, you must enter the server-based directory, C:\servers\data1\datafiles.

If you wish to use any advanced features, such as specifying that the DDFs and data files are not located in the same directory, that the DDFs and data files are located in multiple directories, or modifying the default DBNAME settings, you must click the **Use advanced settings** option.

9 Click **Next**. If you clicked the **Use advanced settings** option, proceed to step 11.

The final wizard step verifies the successful creation of the database.

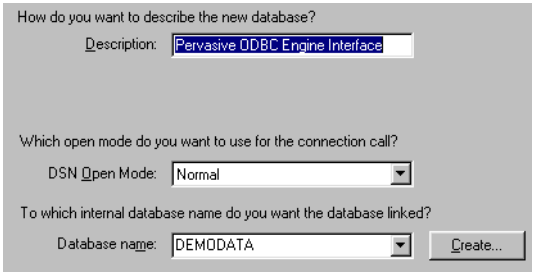
10 Click **Finish**.

The database is ready to access. If you just created a Server DSN for the database engine, you must proceed to setting up client DSNs on each client workstation as explained in "Setting Up Client Access" on page 2-40.

Advanced settings procedure

11 If you clicked the **Use advanced settings** option, you see a window as shown below.

Figure 2-12 Advanced Settings Window


How do you want to describe the new database?
Description:
Pervasive ODBC Engine Interface
Which open mode do you want to use for the connection call?
DSN Open Mode:
Normal
To which internal database name do you want the database linked?
Database name:
DEMODATA
Create...

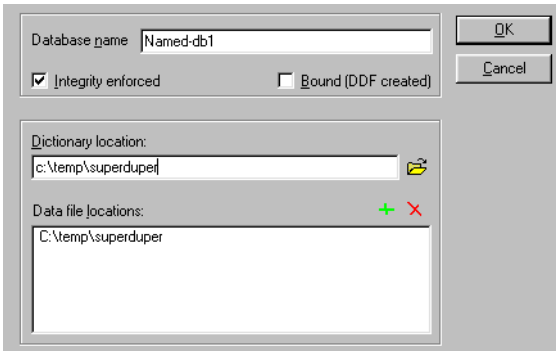
12 In the Advanced Settings window, you can type a description for the new database.

In the **DSN Open Mode** field, choose Normal unless you have reviewed the other options as described in *Advanced Operations Guide*.

In the **Database name** field, choose the DBNAME of the database for which you wish to create the Engine DSN. If you find the DBNAME you want, click **Next** and return to step .

If the DBNAME you want to access is not listed, click the **Create** button to create an internal DBNAME for the database. Click **Next**.

Figure 2-13 Create DBNAME Window



13 In the Database Name field, type in the internal name you want to assign to the database. This name does not need to be the same as the DSN. This name is not displayed in PCC nor is it exposed to users trying to connect to the database during runtime.

For more information in **Integrity enforced** and **Bound**, see "Bound Databases and Enforced Integrity" on page 2-55

Leave **Integrity enforced** checked if you plan to enforce the referential integrity rules defined in the database. Usually it is a good idea to do so.

Click **Bound** (**DDFs created**) if you want to prevent any other databases from being associated with these DDFs and data files. Checking this option creates a new set of empty DDFs. Do not click this option if you are setting up access to an existing database.

In the **Dictionary location** box, type or select the location of the DDFs. This location can be the location of existing DDFs if you are setting up an existing database for use with ODBC, or it can be an empty directory if you are creating a new database.

- For Server databases, this must be a full path name on the server—do not enter or browse a mapped drive path. Browsing for the location only works if you are seated at the server or you are browsing a local directory tree that is identical to the directory tree on the server.
- Workgroup engine users may enter a full path name on a local hard drive, or a mapped drive or UNC name.

In the **Data file location** box, you must accept the default or browse to the location of the data files. This location can be the location of existing data files if you are setting up an existing database for use with ODBC, or it can be an empty directory if you are creating a new database.

- For Server databases, this must be a full path name on the server—do not enter or browse a mapped drive path. Browsing for the location only works if you are seated at the server or you are browsing a local directory tree that is identical to the directory tree on the server.
- Workgroup engine users may enter a full path name on a local hard drive, or a mapped drive or UNC name.

Click **OK** to close the Create Database Name window. This returns you to the Create Database Wizard.

14 Click **Next** and go to step 10.

Setting Up Database Access on a NetWare Server

➤ To Set Up Database Access on a NetWare Server from a Client Workstation

Note This scenario is the only option for NetWare. You must create both the Client and the Engine DSNs from the client workstation using PCC or ODBC Administrator.

If you have not done so already, please review "Prerequisites to Setting up Database Access" on page 2-24 before proceeding.

1 Open Pervasive Control Center (PCC):
Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center

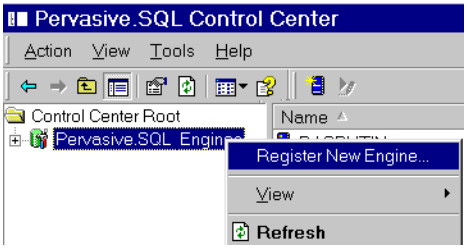
The PCC Namespace contains a list of database engines that have been registered.

2 Double-click on **Pervasive.SQL Engines**.

The PCC Namespace contains a list of database engines that have been registered.

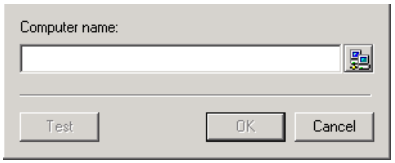
You must register the remote NetWare server with PCC. To do so, right-click on **Pervasive.SQL Engines** then click **Register New Engine**.

Figure 2-14 Registering a New Engine



A dialog box is displayed that allows you to choose the machine name where the Pervasive.SQL server resides.

Figure 2-15 Choosing a Computer Name

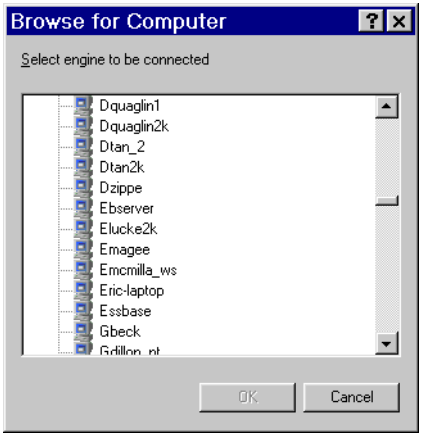


3 Do one of the following:

a. Type the server name in the **Computer name** field and click **OK**.

b. If you are unsure of the server name, click the button located to the right of the name field and browse from the Network list that appears. Click on the desired computer name, then click **OK**.

Figure 2-16 Choosing a Computer Name from a Network List



4 Back in the main screen of PCC, double-click the icon representing the server you are working with. Then double-click the Databases folder associated with your server.

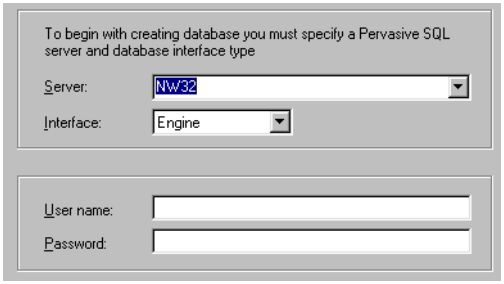
5 Inspect the databases that are listed to determine whether the database you wish to access has already been set up with an Engine DSN.

If the database already appears, then it already has an Engine DSN and you need only set up the client machines with Client DSNs. You are finished with this procedure. Go to "Setting Up Client Access" on page 2-40 for instructions on how to set up the client machines with Client DSNs.

If you do **not** see the name of the database to which you want to connect, continue with step 6.

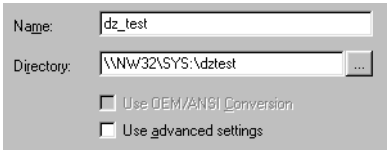
6 Create a New Database for the Server engine:
Right-click on the database folder underneath the Server engine name and select **New Database** from the shortcut menu. The Create Database Wizard appears as shown here.

Figure 2-17 Create an Engine DSN and Connect to a Server Database



7 The Server name should indicate the server that you are currently working with. Choose Engine for the DSN type, and enter your operating system User Name and Password for that server. Remember that you must have administrator-level rights on the server operating system to be able to complete this task. Click **Next** to move to the next screen, shown below.

Figure 2-18 Naming the New Database on the Server



8 In the **Name** field, specify an Engine DSN name for the database. This name will appear in the database listing in PCC after you complete this task.

In the **Directory** field, specify the location of the data files.

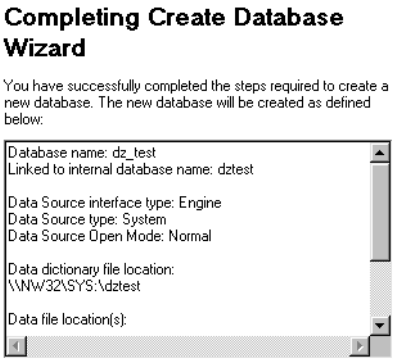
Note The Server engine cannot interpret mapped drive paths. You must specify a full path (not a mapped drive) that makes sense on the server. For example, if you have a mapped drive S:\data1\datafiles on your client, referring to the folder SYS:\servers\data1\datafiles on the server, you must enter the server-based directory, \\servname\SYS:\servers\data1\datafiles.

If you wish to use any advanced features, such as specifying that the DDFs and data files are not located in the same directory, that the DDFs and data files are located in multiple directories, or modifying the default DBNAME settings, you must click the **Use advanced settings** option.

9 Click **Next**. If you clicked the **Use advanced settings** option, proceed to step 11.

The final wizard step verifies the successful creation of the database.

Figure 2-19 Create Database Wizard Complete


Completing Create Database Wizard
You have successfully completed the steps required to create a new database. The new database will be created as defined below:
Database name: dz_test
Linked to internal database name: dztest
Data Source interface type: Engine
Data Source type: System
Data Source Open Mode: Normal
Data dictionary file location:
\\NW32\SYS:\dztest
Data file location(s):

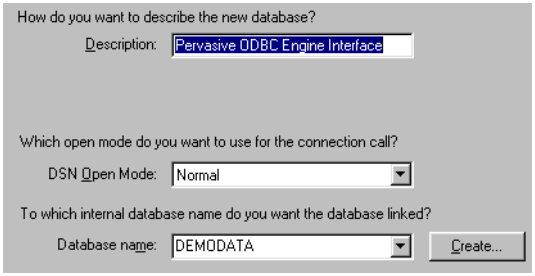
10 Click **Finish**.

The database is ready to access. Proceed to setting up client DSNs as explained in "Setting Up Client Access" on page 2-40.

Advanced settings procedure

11 If you clicked the **Use advanced settings** option, you see a window as shown below.

Figure 2-20 Advanced Settings Window


How do you want to describe the new database?
Description:
Pervasive ODBC Engine Interface
Which open mode do you want to use for the connection call?
DSN Open Mode:
Normal
To which internal database name do you want the database linked?
Database name:
DEMODATA
Create...

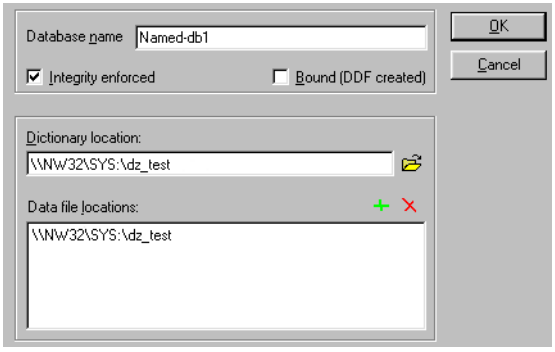
12 In the Advanced Settings window, you can type a description for the new database.

In the **DSN Open Mode** field, choose Normal unless you have reviewed the other options as described in *Advanced Operations Guide*.

In the **Database name** field, choose the DBNAME of the database for which you wish to create the Engine DSN. If you find the DBNAME you want, click **Next** and return to step .

If the database you want to access is not listed, click the **Create** button to create an internal DBNAME for the database. Click **Next** and continue with the next step.

Figure 2-21 Create DBNAME Window



13 In the Create Database Name window, type in the internal name you want to assign to the database. This name does not need to be the same as the DSN. This name is not displayed in PCC nor is it exposed to users trying to connect to the database during runtime.

For more information in **Integrity enforced** and **Bound**, see "Bound Databases and Enforced Integrity" on page 2-55

Leave **Integrity enforced** checked if you plan to enforce the referential integrity rules defined in the database. Usually it is a good idea to do so.

Click **Bound** (**DDFs created**) if you want to prevent any other databases from being associated with these DDFs and data files. Checking this option creates a new set of empty DDFs. Do not click this option if you are setting up access to an existing database.

In the **Dictionary location** box, type or select the location of the DDFs. As previously cautioned, do not enter or browse a mapped drive path because the Server engine cannot interpret mapped drive paths. You must enter a full path name.

Caution Browsing for the location only works if you are using the server itself or you are browsing a local directory tree that is identical to the directory tree on the server. Ensure that you enter a full path name.

In the **Data file location** box, you must accept the default or browse to the location of the data files. As noted previously, browsing for the location only works if you are seated at the server or you are browsing a local directory tree that is identical to the directory tree on the server.

Click **OK** to close the Create Database Name window. This returns you to the Create Database Wizard.

14 Click **Next** and go to step 10.

Setting Up Database Access on a Linux Server

➤ To Set Up a Named Database and Engine DSN from a Linux Server

Database names are created in Linux by using the `dbmaint` utility at the server. For a complete description of `dbmaint`, see "dbmaint" on page 8-18 or read the `dbmaint` man page.

Note This utility can only be run by user accounts belonging to group `pvsw`. See *Getting Started with Pervasive.SQL (Server edition)* for information on Pervasive.SQL Linux utilities and user accounts.

1 To create an empty database, use the following at the command line:

```
dbmaint a | d | l [-b] [-i] [-e] -nDbname
    [-ldictpath] [-ddatapath]
```

The list of commands for dbmaint include:

`a` – add database name
`d` – delete database name
`l` – list all database names

Options include:

`-b` – create Bound database
`-i` – create database with Relational Integrity enforced
`-e` – do not create dictionary files for database
`-n`*DBName* – specify database name
`-l`*Dictpath* – specify dictionary path
`-d`*Datapath* – specify data path
`-a` – show full data in the DBNames list

For example, to create DBName TEST with relational integrity, type:

```
dbmaint a -i -nTEST
```

Note Unless *datapath* is specified, the new database is created in the default location, $PVSW_ROOT/data. Likewise, if *dictpath* is not specified, the dictionary is created in the default location.

- To delete an existing database, use the following at the command line:

  `dbmaint d -n`*Dbname*

  For example, to delete the newly created database TEST, type

  ```
  dbmaint d -nTEST
  ```

- To list all existing databases:

  ```
  dbmaint l [-a]
  ```

2 To set up an Engine DSN, modify the following files:

- `${PVSW_ROOT}/etc/odbc.ini`.

  SQLMGR required settings:

  ```
  [SQLManager]
     MgrPort=1583
     MgrUseTransport=TCP
  ```

- `${PVSW_ROOT}/etc/odbc.ini`

Note The value of `${PVSW_ROOT}` is typically `/usr/local/psql`.

Server data source – the one to which remote calls will be redirected:

```
[DSN name]
   Driver=/usr/local/psql/lib/libsrde.so
   Description=Test Pervasive database
   DBQ=DBName
```

In addition, each data source should be mentioned in the section `[ODBC Data Sources]` as in the following example:

```
[ODBC Data Sources]
   dsnName1=Pervasive.SQL database
   dsnName2=Pervasive.SQL database
```

For example, if you have in `odbc.ini`:

```
[MyDSN]
   Driver=/usr/local/psql/lib/libsrde.so
   Description=test
   DBQ=MyDB
```

then your `odbc.ini` should have:

```
[ODBC Data Sources]
   MyDSN=Pervasive.SQL database
```

Note Because Linux is case sensitive, the `[DSN name]` must be input exactly as listed under `[ODBC Data Sources]`.

- The engine DSN can also be created using the `dsnadd` utility by typing the following at the command line:

  ```
  % dsnadd -dsn=DSNname -db=DBName
  ```

An easy way to verify DBName and DSN configuration settings is to run the supplied odbctest program using the `psql` user account:

```
% /usr/local/psql/bin/odbctest DSN=DEMODATA
```

3 Proceed to setting up client DSNs as explained in "Setting Up Client Access" on page 2-40.

Setting Up Client Access

➢ To make a Pervasive Client able to access a remote Pervasive.SQL database

This procedure explains how to create a Client DSN with the PCC. Client DSNs are used only on client workstations when connecting to a database engine.

Note If you have not done so already, please review "Prerequisites to Setting up Database Access" on page 2-24 before proceeding.

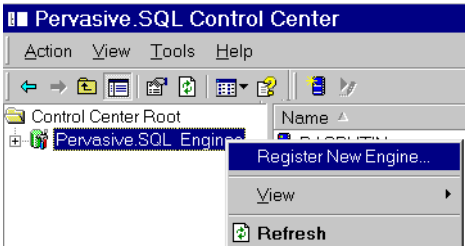
1 Open Pervasive Control Center (PCC):
Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center

2 Double-click on **Pervasive.SQL Engines**.

The PCC Namespace contains a list of database engines that have been registered.

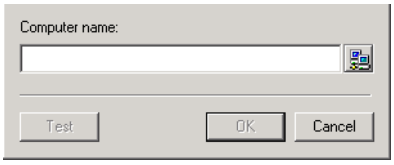
If you do not see the name of the database engine you want to connect to, you must register the remote engine with PCC. To do so, right-click on **Pervasive.SQL Engines** then click **Register New Engine**.

Figure 2-22 Registering a New Engine



A dialog box is displayed that allows you to choose the machine name where the Pervasive.SQL engine resides.

Figure 2-23 Choosing a Computer Name

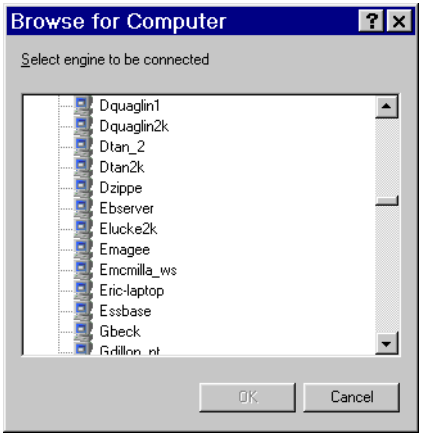


3 Do one of the following:

a. Type the machine name in the **Computer name** field and click **OK**.

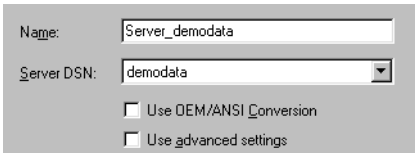
b. If you are unsure of the machine name, click the button located to the right of the name field and browse from the Network list that appears. Click on the desired computer name, then click **OK**.

Figure 2-24 Choosing a Computer Name from a Network List



4 Back in the main screen of PCC, double-click the icon representing the machine you are working with. Right-click the **Databases** folder associated with the machine you have selected. Click **New Database** to start the Create Database Wizard.

Figure 2-25 Create Database Wizard—Client DSN



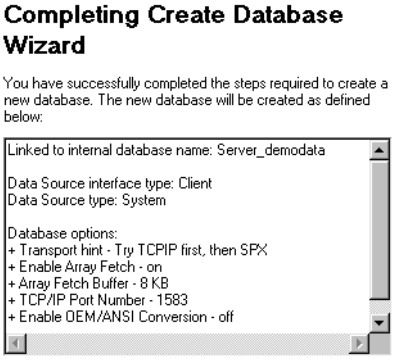
5 In the **Name** field, type a descriptive name for the DSN. This name will be displayed in the ODBC Administrator and to users attempting to connect to ODBC data sources. The maximum length is 32 characters.

In the **Server DSN** field, select the Engine DSN on the machine that you want to associate with the Client DSN you are creating.

If you click **Use advanced settings**, see "Client DSN Options" on page 5-7 in *Advanced Operations Guide* for a discussion of the settings.

6 Click **Next** to display the final confirmation screen.

Figure 2-26 Create Database Wizard—Client DSN Confirmation


Completing Create Database Wizard
You have successfully completed the steps required to create a new database. The new database will be created as defined below:
Linked to internal database name: Server_demodata
Data Source interface type: Client
Data Source type: System
Database options:
+ Transport hint - Try TCPIP first, then SPX
+ Enable Array Fetch - on
+ Array Fetch Buffer - 8 KB
+ TCP/IP Port Number - 1583
+ Enable OEM/ANSI Conversion - off

7 Click **Finish** to complete the wizard.

Note The Client DSN you have just created does not appear in PCC. It appears only in the ODBC Administrator as a System DSN. It also appears in the DSN list presented when local applications attempt to connect to ODBC data sources. Once you have created a Client DSN, you can remove or rename it from the ODBC Administrator, using the **Remove** or **Configure** buttons.

Tip You can set the option to have Client DSNs appear in PCC by right-clicking on the database engine is the list and selecting **View ▸ Show Client Databases**. Once you have set this option, you can remove or configure it using PCC.

➢ Setting Up a Client DSN Using the ODBC Administrator

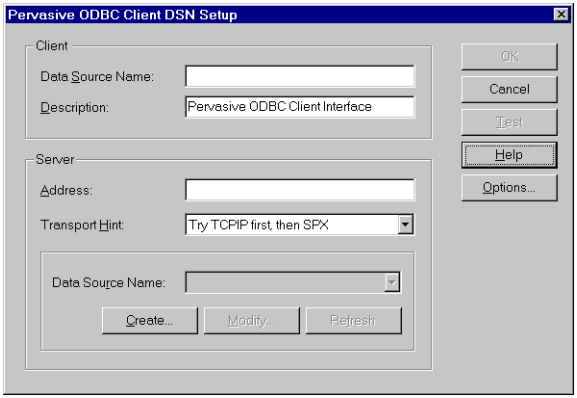
1. Open the ODBC Administrator (from the **Start** menu, select **Programs** ▸ **Pervasive.SQL V8** ▸ **Other Utilities** ▸ **ODBC Administrator**).
2. Click on the **System DSN** tab, then on **Add**.

Note Pervasive.SQL does not support File DSNs. You must use User or System DSNs. System DSNs are generally preferred, because they are available to all users on a given computer.

3. In the Drivers window, select **Pervasive ODBC Client Interface.**

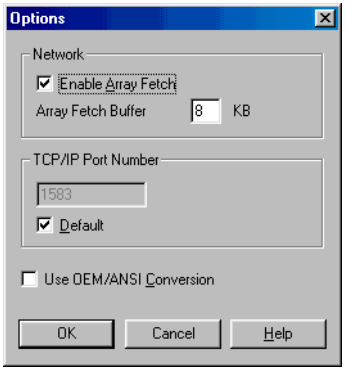
   The following dialog box appears:

Figure 2-27 Pervasive ODBC Client DSN Setup Screen



4. In the Client section, type in a DSN (with a maximum length of 32 characters) for the data source to which you wish to set up a connection. This DSN will help you identify the data source. It will be visible only on the current machine.
5. Type a description of the data source, if desired (with a maximum length of 80 characters).
6. If you want to enable OEM/ANSI conversion, click the **Options** button and make your selection in the dialog box that appears.

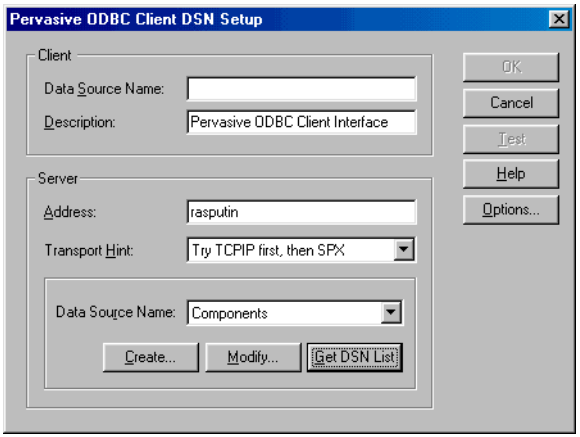
Figure 2-28 Pervasive ODBC Client DSN Options



Do not modify the **Network** settings or **TCP/IP Port Number** unless you have first reviewed the information about these Client DSN options in *Advanced Operations Guide*.

7 Click **OK** to return to the **Pervasive ODBC Client DSN Setup** dialog box.

Figure 2-29 Pervasive ODBC Client DSN Setup Screen #2



8 In the Server area, type in the host name of the computer where the data source resides. You can enter a machine name, TCP/IP address, or an IPX/SPX MAC address.

9 To use an existing database on the server, click the **Get DSN List** button and select the desired DSN from the drop-down list. In the **Server** area, **Data Source Name** refers to an Engine DSN on the server computer.

If no databases appear in the drop-down list, either you selected the wrong server, or you need to have your system administrator name the server databases and create Engine DSNs for each of them before you can access them.

10 Click **OK**.

11 You can now set up another Client DSN or click **OK** to exit the ODBC Administrator.

➤ Setting Up a Client DSN on a Linux Workstation

While it is possible to access a database from a Linux server by a Linux client, there are no Pervasive.SQL utilities (except `dsnadd`) that can be used on the client. A Linux client configuration would be used for independent applications, such as web applications.

To add a client data source, execute the following command:

`dsnadd -dsn=`*`myDSN`* `-desc=`*`datasource`*
`-host=`*`psqlhost`* `-sdsn=`*`svDSN`*

myDSN is a name you want to assign to the new Client DSN.

datasource is any string to describe the data source.

psqlhost is the name of the network host where your Pervasive.SQL database resides.

svDSN is the name of the Engine DSN on the Pervasive.SQL host.

Note The datasource on the server must be named first.

For example, to create a Client DSN named TEST on host NewDev, where the Engine DSN name for the database is NewTest, type

```
dsnadd -dsn=TEST -host=NewDev -sdsn=NewTest
```

For more information about the `dsnadd` utility, please see *Getting Started with Pervasive.SQL* (Server edition).

Accessing Data via ODBC From Other Applications

This section explains how to access data using Microsoft Access and Microsoft Excel.

The examples covered in this section are:

- "Accessing Data Using Microsoft Excel" on page 2-46
- "Accessing Data Using Microsoft Access" on page 2-48

Before You Begin

Does the Database Have a DSN Available?

- If you are connecting from a client workstation or from a Workgroup workstation to a server, you must have a Client DSN defined on your workstation for the given remote database. Information on how to create a Client DSN is provided in "Setting Up Client Access" on page 2-40.
- If you have a Workgroup engine installed on your computer, you may have an Engine DSN defined on your computer for either local or remote databases. Information on how to create an Engine DSN is provided in "Setting Up Database Access on Windows" on page 2-25.

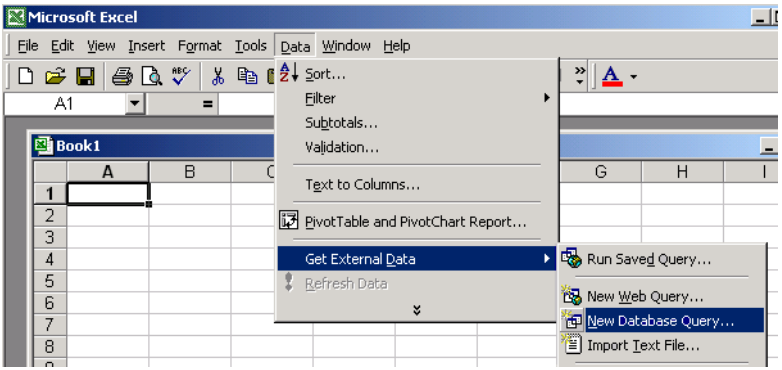
Note The instructions in this section apply only to Pervasive.SQL V8, not to previous versions.

Accessing Data Using Microsoft Excel

➢ To access Pervasive data using Excel

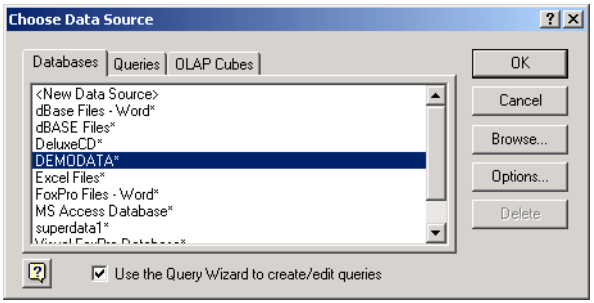
1. You must have the Pervasive.SQL client or any version of the Pervasive.SQL engine installed on the computer where you are using Excel.
2. Start Excel.
3. From the **Data** menu, choose:
   Get External Data ▸ New Database Query as shown below.

Figure 2-30 Accessing Pervasive Data using Microsoft Excel



4 The **Choose Data Source** box lists the defined data sources for any ODBC drivers that are installed on your computer. From this list, click on the Client or Server DSN for the Pervasive database you wish to access, as shown in the example below.

Figure 2-31 Excel Display of ODBC Source List



If the database you want does not appear in the ODBC Source list, see"Before You Begin" on page 2-46.

5 Click **OK**. You may be prompted to login to the Pervasive.SQL database. If the database is not secure, leave the **User** and **Password** fields empty. Otherwise enter your assigned user name and password.

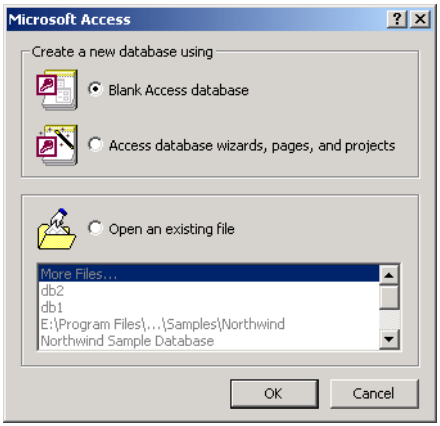
6 The **Query Wizard** opens. Simply follow the wizard to select your options such as which tables to query, how to filter and sort the data, and how you would like Excel to return the Pervasive data to you for your use.

Accessing Data Using Microsoft Access

➢ To access data from Microsoft Access

1. Open Microsoft Access.
2. From the Access dialog box, choose **Blank Access database** as shown below. Click **OK**. (Note that you may also add Pervasive.SQL tables to an existing Access database.)

Figure 2-32 Create a New Database using Microsoft Access

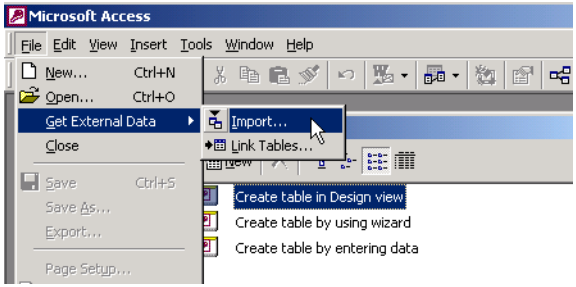


3. Next, the **File New Database** dialog box opens and asks you to name the new database. Name the database and click **Create**.
4. From the Access menu, choose:
   File ▸ Get External Data ▸ Link Tables.

Note You have the option to **Import** data or **Link Tables** to the new database. When you choose **Import**, you break the link to the ODBC data source immediately following the import procedure. Essentially, **Import** creates a static copy of the data. When you choose **Link Tables**, Microsoft Access keeps the connection open and remains dependent upon the ODBC data source each time the data is accessed. This way, the data you see reflects any changes to the data at its source.

Note If you wish to link to a file on a local area network, make sure to use a universal naming convention (UNC) path, instead of relying on the drive letter of a mapped network drive in Windows Explorer. A drive letter can vary on a computer or may not always be defined, whereas a UNC path is a reliable and consistent way for Microsoft Access to locate the data source that contains the linked table.

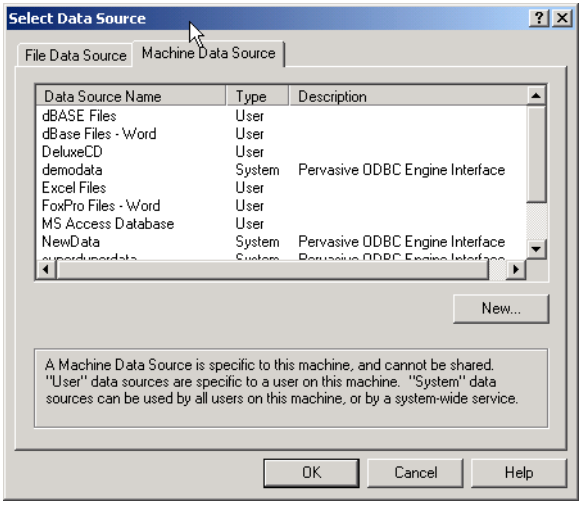
Figure 2-33 Importing External Data Using Access



5 In the **Link** dialog box, in the **Files Of Type** box, select **ODBC Databases.**

6 The **Select Data Source** box lists the defined data sources for any ODBC drivers that are installed on your computer. Click on the **Machine Data Source** tab as shown in the figure below.

Figure 2-34 Access Display of ODBC Source List


Select Data Source
File Data Source
Machine Data Source
Data Source Name Type Description
dBASE Files User
dBase Files - Word User
DeluxeCD User
demodata System Pervasive ODBC Engine Interface
Excel Files User
FoxPro Files - Word User
MS Access Database User
NewData System Pervasive ODBC Engine Interface
New...
A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.
OK
Cancel
Help

7 Select the ODBC data source that you want to link. If the ODBC data source that you selected requires you to log on, enter your user name and password (additional information might also be required), and then click **OK**.

Note To define a new data source for any installed ODBC driver, click **New**, and then follow the instructions in the **Create New Data Source** dialog box and the dialog boxes that follow it before proceeding.

Tip If you are linking a table, select the **Save The Login ID And Password** check box to store the information for the table in the current database, so that users will not have to enter it each time. If you leave the check box cleared, all users must enter the logon ID and password every time they open the table with Microsoft Access in each new session. Your network administrator can also choose to disable this check box, requiring all users to enter a user name and password each time they connect to the database.

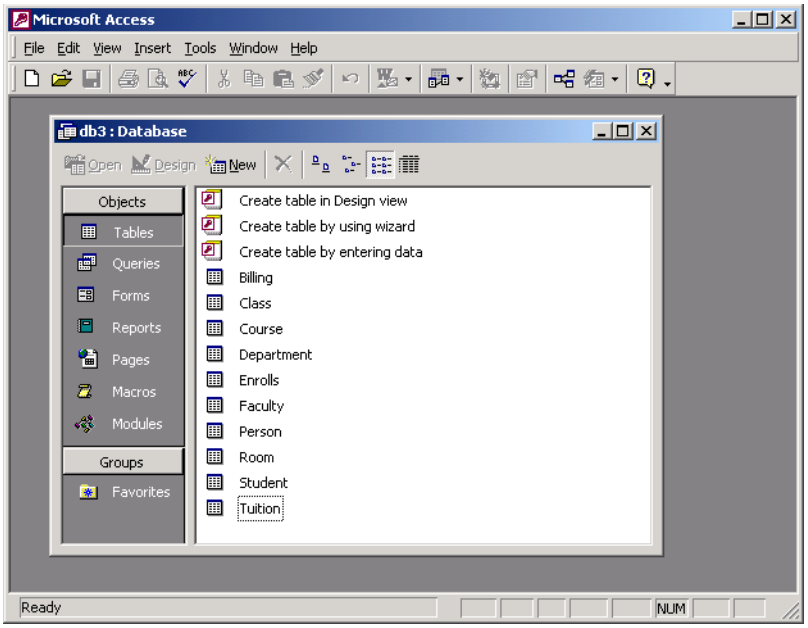
If the database you want does not appear in the ODBC Source list, see"Before You Begin" on page 2-46.

8 The Access **Link Tables** dialog box opens. Click each table that you want to import or link, and then click **OK**.

Note Microsoft Access cannot display more than 256 columns in a table. If you need to display more than 256 columns, you may wish to use a different tool.

9 Linking to your Pervasive data is complete. As shown in the figure below, Access presents you with options for designing the new database. View the linked tables by double-clicking on the table name.

Figure 2-35 Using Pervasive Data in Microsoft Access



Note If you are linking a table and it does not have an index that uniquely identifies each record, then Microsoft Access displays a list of the fields in the linked table. Click a field or a combination of fields that will uniquely identify each record, and then click **OK**.

Deleting DSNs

This section describes how to delete a DSN. One reason you may have to perform this task is if you are upgrading from Pervasive.SQL 7. If this is the case, you must delete your existing DSNs before installing Pervasive.SQL V8 and re-create them using Pervasive.SQL V8.

This procedure does not delete Data Dictionary Files (DDFs) or data files.

➤ To Delete a DSN using ODBC Administrator (for Windows DSNs only)

1. From the **Start** menu, choose **Programs ▸ Pervasive.SQL V8 ▸ Other Utilities ▸ ODBC Administrator**.
2. In the **ODBC Administrator** window, click on the **System DSN** tab.
3. Click on the DSN you wish to remove, and click **Remove**. You are prompted to confirm removal of the DSN. Click **Yes**.
4. After the DSN has been removed, click **OK** to exit ODBC Administrator.
5. If you are simply deleting an unwanted DSN, you are finished. If you need to re-create the DSN, you should refer to one or more of the following sections:

If you need to do this	... refer to this section:
Re-create an Engine DSN on a Server engine or a Workgroup engine	One of: ◆ "Setting Up Database Access on Windows" on page 2-25 ◆ "Setting Up Database Access on a NetWare Server" on page 2-31 ◆ "Setting Up Database Access on a Linux Server" on page 2-37
Re-create a Client DSN on a client workstation	"Setting Up Client Access" on page 2-40

➤ To Delete a DSN Using PCC

1. From the **Start** menu, choose **Programs ▸ Pervasive.SQL V8 ▸ Control Center**.
2. Double-click **Pervasive.SQL Engines**.

   If you wish to remove an Engine DSN on the server, you must have administrative rights on the server to do so. If you do not see the name of the server computer listed in the left-hand window of PCC, you must register the server. See "Registering or Removing a Server" on page 3-4.
3. Double-click the icon representing the computer where you want to remove the DSN. Double-click **Databases**.
4. Right-click on the database you wish to remove. Choose **Delete** from the pop-up menu.
5. In the Drop Database Wizard that appears, clear **Delete database name**.

Caution Be certain **Delete database name** is not checked. If this box is checked, the internal database name as well as the DSN will be deleted, and you will have to re-create the internal database name.

Also, if you want to save the system files and DDFs, make sure the **Delete system files** is not checked. One reason that you may need to preserve these files is because a Btrieve application is accessing them.

6. Click **Next** to display the confirmation screen and then click **Finish**. You will notified of a successful deletion. Click **Close** to exit the **Delete Database Wizard**.

7 If you are simply deleting an unwanted DSN, you are finished. If you need to re-create the DSN, you need to select one or more of the following options:

If you need to do this	... refer to this section:
Re-create an Engine DSN on a Server engine or on a Workgroup engine	One of: ◆ "Setting Up Database Access on Windows" on page 2-25 ◆ "Setting Up Database Access on a NetWare Server" on page 2-31 ◆ "Setting Up Database Access on a Linux Server" on page 2-37
Re-create a Client DSN on a client workstation	"Setting Up Client Access" on page 2-40

Bound Databases and Enforced Integrity

When creating a database, you are offered the options of **Integrity Enforced** and **Bound (DDFs created)**.

- **Integrity Enforced** means it enforces any triggers and referential integrity defined within the database. Generally, the only time you would *not* want this option checked is when doing a large download of data from an existing database.
- A **Bound** database associates a database name with a single set of Data Dictionary Files (DDFs), which refer to only one set of data files. If DDFs are bound, you can't use those DDFs for several databases, nor can you refer to the data files by more than one set of DDFs. This protects the database from tampering. If you delete a bound database, the associated DDFs are also deleted.

Binding a database ensures that the MicroKernel enforces the database's defined security, referential integrity (RI), and triggers, regardless of the method you use to access the data.

Even if you do not bind your database, Pervasive.SQL automatically stamps a data file as bound if has a trigger, has a foreign key, or has a primary key that is referenced by a foreign key. Thus, a data file may be part of an unbound database, but be bound. In such cases, the MicroKernel enforces integrity constraints on the file as if it were part of a bound database.

When you create a bound database or bind an existing database, Pervasive.SQL stamps every dictionary and data file with the name of the bound database. Also, Pervasive.SQL stamps every data file with the name of the table associated with that data file. In addition, when you add new tables or dictionary files to the bound database, Pervasive.SQL automatically binds them.

Caution Under certain conditions related to relational database structures, access to data through the Btrieve interface is restricted. If you plan to use both SQL/ODBC and Btrieve to access your data, and your database is bound or uses triggers, stored procedures, security, or referential integrity, please refer to "Interactions Between Btrieve and Relational Constraints" on page 6-9 in *Advanced Operations Guide*.

Using the Fast User Switching Feature of Windows XP

Fast user switching is a feature of Windows XP Home Edition and Windows XP Professional that allows you to switch between users without logging off from the computer. Multiple users can share a computer, switching back and forth among users without closing the programs each user is running. The users are all local to the computer, not logged in via a network. Only one user at a time can use the computer interactively.

(The Pervasive.SQL Server engine is not supported on Windows XP Professional or Home Edition. This support will be available when Windows XP Server is released.)

As of this release of Pervasive.SQL, the following Microsoft restrictions apply when fast user switching is turned on. These are restrictions of the operating system.

- The computer cannot be logged on to a network domain.
- The Microsoft Client for Netware cannot be installed.
- The Novell Client for Netware cannot be installed.
- The Serial Keys accessibility feature will not work.
- Offline files, such as Windows XP Offline Documentation, must be disabled.

Fast user switching allows only two types of users, classified as administrators or limited. Only administrators can turn on or turn off fast user switching.

Windows XP Professional allows two modes of operation, local and remote desktop. Remote desktop uses an XP client to access an XP machine from a remote computer. In many respects, the remote desktop feature is similar to terminal services under Windows NT or Windows 2000.

(You may also use a Pervasive.SQL client on an XP machine to communicate with a Pervasive.SQL Server engine across a network. The client functions the same as it does on any other Windows platform supported by Pervasive.SQL.)

Fast User Switching in Local Mode with Pervasive.SQL

The following conditions apply when you use a Pervasive.SQL client and Workgroup engine in local mode. Local mode refers to a local client communicating with a local engine.

- Only one instance of the engine may be run at a time. You cannot run separate copies of the engine within separate user sessions. If you attempt to start the engine that has already been started by another user, the engine will not restart. No error message appears and no tray icon appears.
  - If started by a user, the engine runs in the session of the first user to start the engine. The operating system rights of the first user to start the engine determine access rights to the database files. For example, a "limited" user can start the Pervasive.SQL engine but cannot create a new database.
  - If the Pervasive.SQL Workgroup engine is started as a service, the operating system rights of the System Account determine access rights to the database files. See “Running the Workgroup Engine as a Service” on page 10-15 in *Getting Started with Pervasive.SQL* (*Workgroup* edition).
- Changes made to a database in one session are available to users of the database in other sessions. For example, User A adds a record to Database 1. User B fast user switches to her session. User B sees the record in Database 1 added by User A.
- A database accessed by one user’s session can lock the database from other users’ sessions. For example, User A has Database 1 open in the PCC. User B fast user switches to her session and attempts to add security to Database 1. User B is prevented from adding security to the database.

Caution If you run the Workgroup engine as a console application, the first user to start the engine should not stop the engine if other users are accessing the engine. In addition, the first user must not log off because this causes the engine to terminate.

As an alternative to running the engine as a console application, you may run the engine as a service. See “Running the Workgroup Engine as a Service” in *Getting Started with Pervasive.SQL* (*Workgroup* edition).

chapter

Using the Pervasive Control Center

3

A Brief Tour of Pervasive Control Center

Pervasive Control Center (PCC) is an easy-to-use, graphical tool designed to help you create and manipulate databases and control your DBMS. It allows you to access nearly all the functions of the product from one place. This chapter leads you on a tour of PCC to help you learn the interface, the variety of Pervasive.SQL V8 tools, and common operations launched from PCC.

The topics in this chapter include:

- "An Overview of Pervasive Control Center" on page 3-2
- "Registering or Removing a Server" on page 3-4
- "Viewing Database Engines" on page 3-6
- "Pervasive Control Center Wizards" on page 3-8
  - "Adding or Creating a Database" on page 3-10
  - "Deleting a Database" on page 3-13
  - "Adding a Table" on page 3-15
  - "Dropping a Table" on page 3-20

PCC "Shortcuts"

- "Stopping and Restarting Services on Windows Servers" on page 3-28

PCC Functions

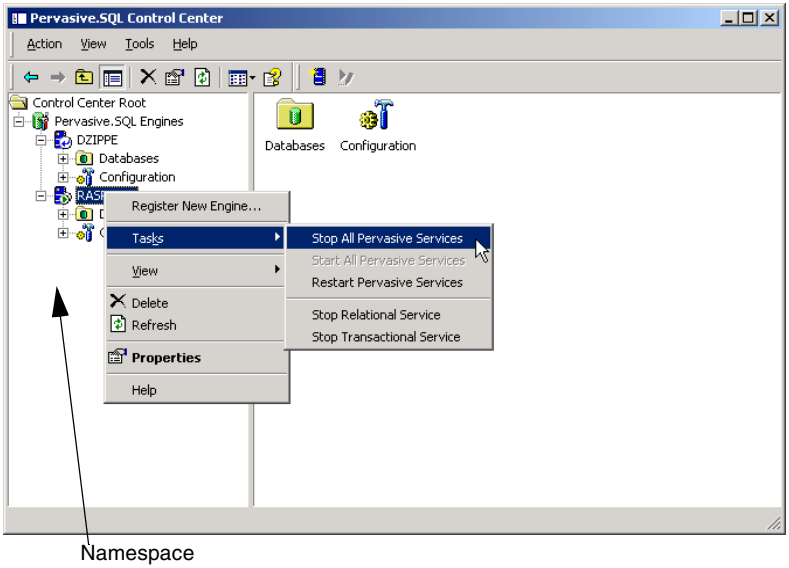
- "Viewing and Modifying Table Properties" on page 3-29
- "Viewing and Modifying Data" on page 3-32
- "Setting Database Security" on page 3-22
- "Exporting/Importing Data" on page 3-33
- "Checking Consistency and Referential Integrity" on page 3-42

An Overview of Pervasive Control Center

Pervasive Control Center (PCC) is an integrated framework in which users can connect to Pervasive.SQL engines, set up and modify databases, query and update data, and tune engine performance.

PCC uses a Windows Explorer-like motif—a tree of objects—referred to elsewhere as the "Namespace." This tree of objects can be opened or expanded to reveal more detail. Examples of objects include engines, databases, tables, users and engine configuration settings. Figure 3-1 shows a picture of PCC.

Figure 3-1 Pervasive Control Center



Each object in the tree, when selected, has its own set of functions and its own set of tools to enable these functions. For example, if one of the **Databases** located under a Pervasive.SQL engine is selected, a different set of tools appears in the toolbar than if **Configuration** is selected. The **Configuration** tools function separately from the Pervasive.SQL engine tools.

Much of the functionality of PCC is implemented as wizards that act upon objects in the namespace. To perform a function on a given object, right-click on the object and choose from the list of actions in the **Tasks** menu.

Some utilities have not yet been tightly integrated within the PCC framework. However, they may still be started from within PCC by selecting them through the **Tools** menu, which is available when anything but the Control Center Root is selected. The non-integrated tools that may be selected this way include:

- Function Executor
- Pervasive System Analyzer
- Gateway Locator (Workgroup only)
- Maintenance
- Monitor
- Rebuild
- License Administrator Utility
- Custom (third-party) tools

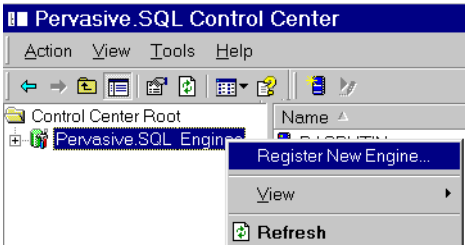
Registering or Removing a Server

You can use Pervasive Control Center (PCC) to work with database engines that are on your machine or with remote server engines. To work with a remote server engine, you must introduce it to PCC. This procedure is called *registering* the server.

➤ To register a remote server with PCC

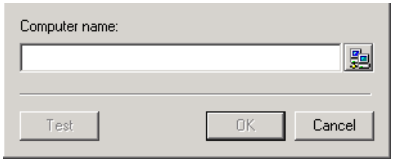
1. Open PCC:
   Choose **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center.**
2. Within PCC, double-click on **Pervasive.SQL Engines** to see a list of engines that are already registered.
3. Right-click on **Pervasive.SQL Engines** and select **Register New Engine**. Type in or choose the server you want to connect to.

Figure 3-2 Registering a New Engine



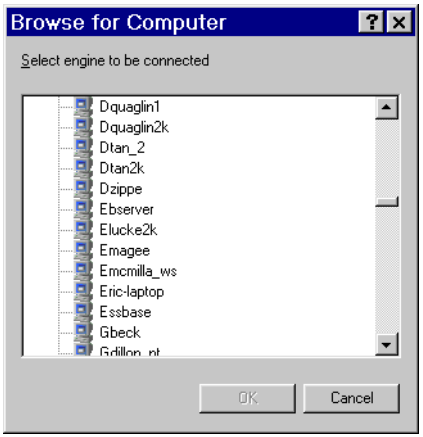
A dialog is displayed that allows you to choose the machine name where the Pervasive.SQL server resides.

Figure 3-3 Choosing a Computer Name



4. If you are unsure of the server name, click the button located to the right of the name field and browse from the Network list that appears, as shown below.

Figure 3-4 Choosing a Computer Name from a Network List



5 Select the computer name from the list or enter the computer name in the field and click **OK**.

The server should now appear in the Namespace window of PCC.

Note If you have engines registered that are not running, you may encounter delays in Configuration as PCC periodically attempts to contact these engines. To eliminate the delay, unregister the "dead" engines. You can minimize the delay by performing the following procedure:

In PCC, right-click **Pervasive.SQL Engines** and choose **Properties**. Set **Poll interval** (**seconds**) to 999. Click **OK**.

➤ To remove a remote server from the Namespace

This procedure does not erase the data or database from the server. It only keeps the server from appearing in PCC on your computer.

1 Open PCC:
Choose **Start** ▸ **Programs** ▸ **Pervasive.SQL V8** ▸ **Control Center.**

2 Within PCC, double-click on **Pervasive.SQL Engines**.

3 Right-click on the server you want to remove and select **Delete**. PCC disconnects from the server, and it is removed from the Namespace.

Viewing Database Engines

Pervasive Control Center (PCC) provides a unique view of available database engines, databases, and actual data. This section explains how to view these objects in PCC.

➤ To view registered database engines

1. Open PCC:
   Choose **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center.**
2. You can view the database engines that you have registered in PCC by double-clicking **Pervasive.SQL Engines** in the left-hand pane (the Namespace) of PCC.

   If you want to add more database engines to the list, follow the instructions provided in "Registering or Removing a Server" on page 3-4.

   By right-clicking on a particular database engine, you can choose to view its properties, refresh its database list, or delete it.

Note If you have engines registered that are not running, you may encounter delays in PCC as it periodically attempts to contact these engines. To eliminate the delay, unregister the "dead" engines. You can minimize the delay by performing the following procedure:

In PCC, right-click **Pervasive.SQL Engines** and choose **Properties**. Set **Poll interval (seconds)** to 999. Click **OK**.

Interpreting Engine Status Icons

Computer names are listed in the namespace tree under the Pervasive.SQL Engines node. An icon indicating the state of the Pervasive.SQL engine on that computer accompanies each engine

name. The possible states of the database engine are outlined in the following table.

Table 3-1 Pervasive.SQL Engine States

Engine Status	Small Icon	Large Icon
Workgroup (local or remote)		
Windows NT/2000 Server (Green triangle) (relational and transactional)		
Windows NT/2000 Server (Green triangle) (transactional only)		
NetWare Server (Blue triangle)		
Linux Server (Brown triangle)		
Engine stopped		
Engine detection in progress		
No Service Found		

Note When PCC is next started, engines remain in the Namespace even if the engines are not running. Engines that are not running are listed in the namespace with an appropriate icon indicating the engine is not running.

Pervasive Control Center Wizards

This section provides information about wizards available from Pervasive Control Center (PCC). These wizards can be invoked using a **DOS command line** or through a window in PCC. In this manual, PCC is the default starting point for creating and deleting tables, data and databases.

If you invoke a wizard from the command line (see Figure 3-5), a login screen requires you to specify a **Server** and **Database** to use. If the database you want to access is secure, the login screen also requires you to enter a **User Name** and **Password**. The login screen also appears when creating a new database. The following screen is an example of a wizard invocation using a DOS command line:

Figure 3-5 DOS Command Line Screen Example

The following table provides the file names of the wizards if you wish to start them from the command line:

Table 3-2 File Names of Wizards

File Name	Wizard
createdb.exe	Create Database
checkdb.exe	Check Database
crtblwzd.exe	Create Table
dropdb.exe	Drop Database
droptab.exe	Drop Table
expwizrd.exe	Data Export
impwizrd.exe	Data Import

Table 3-2 File Names of Wizards continued

File Name	Wizard
nulcnvwz.exe	Null Conversion
psawizrd.exe	Pervasive System Analyzer

Adding or Creating a Database

To add or create a database, use the Create Database Wizard. Newly created databases are empty and may be populated with tables using the Add Table Wizard described in "Adding a Table" on page 3-15. (You may also create a database with Table Designer. See "Table Designer" on page 5-1.)

You also use the Create Database Wizard to create Engine and Client DSNs for pre-existing databases. This allows a database to be accessed using ODBC. For information on how to do this, see "Setting Up ODBC Database Access" on page 2-21.

Note If you wish to add a database to a Server engine, you must have administrative rights on the server operating system. If you do not have administrative rights, you will not be permitted to add the database.

➤ To add or create a database

1. Within PCC, right-click on **Databases** under the machine on which you want to create the database.

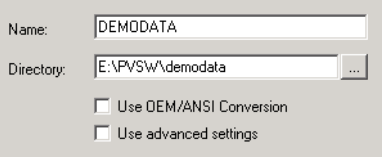
   If you do not see the name of the machine where you want to create the database, see "Registering or Removing a Server" on page 3-4.

2. Select **New Database** from the shortcut menu.

3. On a Server engine, you may be prompted to login. If so, supply a user name and password for the server operating system.

   The Create Database Wizard starts with the following dialog box:

Figure 3-6 Create Database Wizard Dialog Box



Enter the name of the new database and the location where its files will reside.

Note When creating a database on a remote machine, remember that the directory entered in this dialog box must be valid as if it was entered on the remote machine. If you are working with a Server engine, you must specify a full path (not a mapped drive) that makes sense on the server. Optionally, you may use UNC paths:

For NetWare systems, use the following path:
`\\server\vol1:\path`

For Linux systems (when Samba is not used), use the following path:
`\\server\$PVSW$\[path on Linux from root '\' dir]`

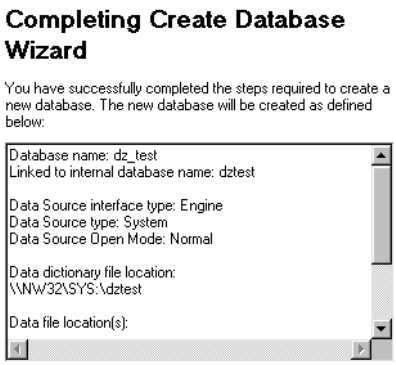
For Windows (or Linux systems with Samba), use the following path:
`\\server\sharename\path`

Optionally, you may check the **Use advanced settings** box to specify more details on how the new database should be created, or if you want to create DDFs for existing data files. If this option is selected, an additional dialog box appears later. This option is not required for most database creation tasks. For instructions on using the advanced settings, see "Advanced settings procedure" on page 2-28.

Click **Next** when done.

4 Now that all of the required information has been gathered, the Create Database Wizard presents a screen indicating the actions it is about to take. This screen allows you to confirm the settings before they are committed to the engine. Click **Finish**.

Figure 3-7 Create Database Wizard - Complete Dialog Box


Completing Create Database Wizard
You have successfully completed the steps required to create a new database. The new database will be created as defined below:
Database name: dz_test
Linked to internal database name: dztest
Data Source interface type: Engine
Data Source type: System
Data Source Open Mode: Normal
Data dictionary file location:
\\NW32\SYS:\dztest
Data file location(s):

5 If no errors have occurred, the wizard indicates success and displays a final dialog box (not shown). Click **Close**.

Now, if you look in the PCC namespace, a new entry should appear for the database just created. You may have to refresh the list in the Pervasive Control Center for the new database to show. To do this, point the cursor at the menu, click **Action**, then **Refresh.**

Note If you have created a brand new database, where no DDFs or data files existed previously, your new database is empty. No tables or columns have been created. You can use the Create Table Wizard, described in "Adding a Table" on page 3-15, to define tables and columns.

Deleting a Database

PCC includes a Drop (delete) Database Wizard. You can use the wizard to delete a DSN, a Database Name, or an entire database. (You may also delete a database with Table Designer. See "Table Designer" on page 5-1.)

To delete an Engine DSN, a Database Name, or a complete database from a server, you must have administrative rights on the server.

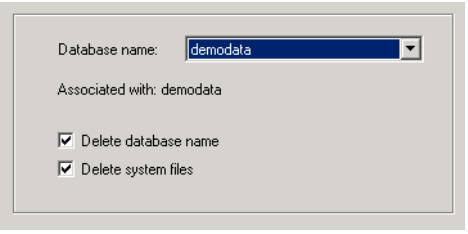
Caution This procedure is different than removing an engine from the PCC Namespace. This procedure may permanently delete data files and data dictionaries.

Caution Do not use the Drop Database Wizard (or the PCC Delete command) to delete DataExchange-enabled databases or DataExchange templates. Use the Deactivation Wizard to remove DataExchange-enabled databases and the Template Remover Wizard to remove templates. The Deactivation Wizard and the Template Remover Wizard can be accessed from the PCC after the Pervasive.DataExchange product is installed.

➤ To remove an existing database from an engine

1. In PCC, right-click on the database to be deleted.
2. Select **Delete** from the shortcut menu. The following dialog box appears:

Figure 3-8 Drop Database Wizard Dialog Box



This dialog provides options to customize how much of the database to remove from the system. By default the database DSN, the internal database name and all data files are removed.

If no options are selected, only the DSN is deleted. The internal database name, the DDFs, and the data files are not affected.

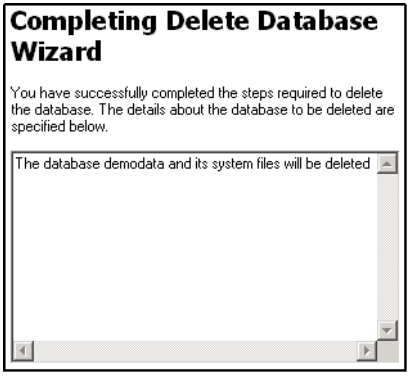
If **Delete database name** is the only option checked, the DSN and the internal database name are deleted, but the DDFs files and the data files are not affected.

If **Delete system files** is checked, the DDFs and data files are deleted. You cannot select this option without also selecting **Delete database name**.

Caution Depending on the options you have selected, deleting a database may erase the related data files and data dictionaries from your hard drive. Do not delete a database unless you are certain you want to do so.

3 Click **Next**. To delete the database, click **Finish**. To quit without deleting the database, click **Cancel**.

Figure 3-9 Completing Drop Database Wizard Dialog Box


Completing Delete Database Wizard
You have successfully completed the steps required to delete the database. The details about the database to be deleted are specified below.
The database demodata and its system files will be deleted

4 After you click **Finish,** the wizard displays a final dialog box (not shown) that indicates success. Click **Close**. At this point the database should disappear from the PCC Namespace. You may need to refresh the screen in the PCC to view this change. To do this, go to the toolbar, click **Action**, then **Refresh.**

Adding a Table

Tables are the objects in which databases store data. PCC's Create Table Wizard enables you to easily add and remove tables from any existing database. (You may also add a table with Table Designer. See "Table Designer" on page 5-1.)

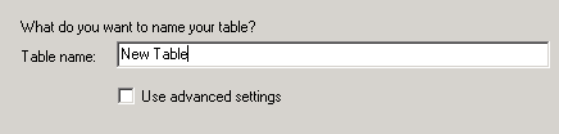
Note To create a table in a database, database security must be turned off or you must have access rights to create tables.

Caution It is highly recommended that you backup all your DDFs and data files before you perform functions through Table Designer. That tool lets you modify your database schema and data. If you inadvertently set the options incorrectly or enter incorrect data, you could change your files in an irreversible manner. Full recovery will be possible if you have performed a backup.

➤ To create a new table within a database

1. Click on the **Tables** Namespace node beneath the database in which the new table is to be created.
2. Right-click on the background in the table list in the right pane of the Control Center window or right-click on the "Tables" icon underneath a given database icon.
3. Select **New Table** from the shortcut menu. The following dialog box appears:

Figure 3-10 Create Table Wizard Dialog Box

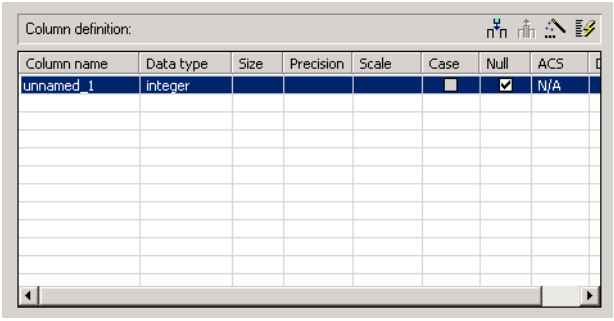


Enter the name of the new table. Click **Next** when done.

For advanced users only: you may select the **Use advanced settings** box in order to create a table definition for an existing Btrieve file. The steps to use this advanced option are described in *Advanced Operations Guide*, "Adding Relational Access to Btrieve Files" on page 13-1.

4 The columns that are to appear in the new table must now be defined. The following dialog box presents a grid display in which columns may be defined:

Figure 3-11 Create Table Wizard - Designer View Dialog Box



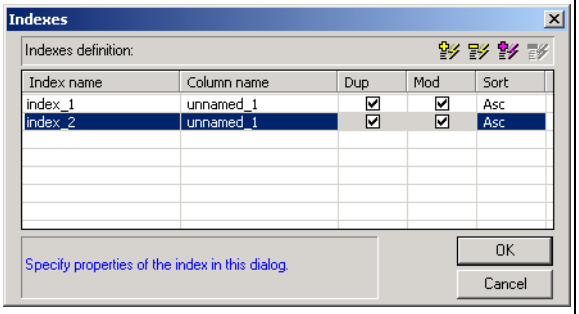
Create Table Wizard Design Buttons

Use the design buttons (located at the right of the window) for ease in creating a table. Their functions are described in Table 3-3.

Table 3-3 Table Wizard Tools

Function	Icon
Add Column - Click this button to add a column.	
Remove Column - Highlight the column to be removed and click this button.	

Table 3-3 Table Wizard Tools

Function	Icon
Index Window - Click this button for a dialog box in which you can specify properties of the index.	 
Index Window Tools: **Add Index** **Remove Index** **Add Segment** **Remove Segment**	

Create Table Wizard Column Attributes

Name
Enter an alpha-numeric name for the column. The limit is 20 characters. You can have spaces in the name, but if you have spaces, you must always enclose the name in quotes whenever you are referring to it in SQL statements. Avoiding the use of spaces is recommended.

Data Type
Choose a data type for the column.

Size
For character or binary fields, specify how many bytes are permitted in this field.

Precision
For floating point values, specify the number of significant digits.

Scale
For floating point values, specify the number of significant digits that are to the right of the decimal point.

Case
For character data types, checking the **Case** attribute forces the database engine to use case-insensitive comparisons when searching for values in the database.

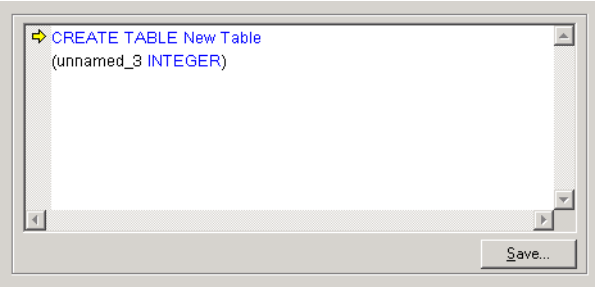
Null
For data types that allow NULL values (IDENTITY and BIT data types cannot be null), checking the **Null** attribute permits NULL values in the column.

ACS
If you wish to use an alternate collating sequence (ACS) for this column, specify the path name to the ACS here.

Default
If you wish to specify a default value for this column, enter the value here. The default value is used if a row is inserted and no value is provided for this column.

5 The next dialog box (Figure 3-12) displays the SQL statement that could be used to recreate this table at a later date. This statement can prove useful in replicating a database structure at another location, or as a safety measure in case the table must be re-created.

Figure 3-12 Create Table Wizard - Generate SQL Script Dialog Box



6 You can click the **Save** button to save the script so that it can be run in the Pervasive.SQL data manager at any time. Click **Next**.

7 By clicking **Finish** in the dialog box below (Figure 3-13), the changes that have been entered thus far will be committed to the database.

Figure 3-13 Create Table Wizard - Complete Dialog Box

Completing Create Table Wizard
You have successfully completed the steps necessary to create the table.
When you click the Finish button, the table will be created.
To create the table, click Finish.

8 The wizard displays a final dialog box (not shown) indicating that the operation was a success. Click **Close**.

Creating Tables for Existing Data Files

Sometimes a data file already exists, but it has no table definition allowing it to be accessed with ODBC. The advanced settings of the Create Table Wizard allow for adding table definitions for existing data files. For detailed instructions on how to perform this complex procedure, see *Advanced Operations Guide*.

Dropping a Table

Dropping a table using the PCC's Drop Table Wizard is very similar to dropping a database. (You may also delete a table with Table Designer. See "Table Designer" on page 5-1.)

Note To delete a table from a database, database security must be turned off or you must have access rights to delete tables. See Chapter 2—"Using the Pervasive Control Center," for information on access rights.

Caution This procedure may permanently delete data files and DDFs.

➤ To delete an existing table from a database

1. Click on the **Tables** Namespace node under the database whose table you want to drop.
2. Right-click on the table to be dropped in the table list pane on the right side of the Control Center window.
3. Select **Delete** from the shortcut menu.
4. The following dialog box appears, allowing you to delete the table.

Figure 3-14 Drop Table Wizard

The option **Delete Data file** not only removes the table from the database but also removes the actual data file itself from the machine's hard drive. Click **Finish**.

5 The final **Drop Table Wizard** dialog box (not shown) indicates to you that the table was successfully dropped. Click **Close**.

Note When a table is in use, it cannot be dropped. If you are unable to drop a table, use Monitor to verify whether the table is in use.

Setting Database Security

For further information on database security, see *Advanced Operations Guide*, "Owner Names and Relational Security."

Database security helps prevent intruders from accessing the data on your databases or unintentional damage to data by authorized users. By default, security is turned off. Security is enabled on Pervasive.SQL databases through the PCC or by executing an SQL statement.

Note When turning security on or off, all database connections must be closed. Because SQL Data Manager uses additional connections, you must close all SQL Data Manager windows before attempting to turn security on or off. If you are viewing the contents or properties of any table in the database, then you must close that window before proceeding.

Turning Security On and Off

➢ To set database security from within the PCC

1. Right-click on a database icon in PCC Namespace. Select the **Properties** menu item.
2. Select the **Security** tab in the Database properties dialog box.

Figure 3-15 Database Properties Dialog Box

3. Enter a password for the Master user.

Caution Be sure to specify a password with significant length, at least five characters. Do not leave the password field blank because doing so creates a major security risk for your database.

Click **OK** when done.

The database is now secured, and a single user named "Master" has been created with the password you specified. Until additional users are created and assigned permissions in the database, only the Master user is permitted to view or update data.

A secure database supports individual users and groups of users who have the same set of permissions. Additional users and groups may be added to the database through the **Users** namespace node under the database. For more information about adding users and groups, see "Working with Groups and Users" on page 3-23.

➤ To turn off database security

Caution Turning off database security deletes all users, groups, and permissions. If you turn security back on, you must re-create all users and groups.

1. Right-click on the icon for your database in the PCC Namespace and choose **Properties** in the shortcut menu.
2. Select the **Security** tab in the Database properties dialog box.
3. Since security is on, the pane includes a check box entry to turn security off. Select it and press **OK** to turn off security. Unless you have an open database session, Pervasive.SQL prompts you for the Master password before turning off security.

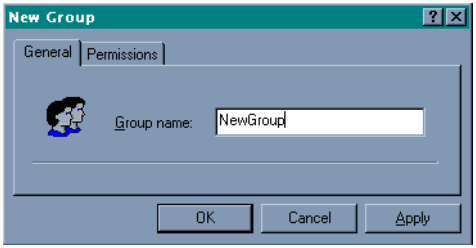
Working with Groups and Users

For any data files in your database with Btrieve owner names set, you cannot use PCC to grant access to these files. You must use the GRANT statement with the appropriate owner name. Refer to the *SQL Engine Reference* for a discussion of the GRANT statement.

➤ To add a new group to the database

1. Select the Namespace node **Users** under the database in which you would like to add a new group.
2. Right-click **Users**. Select **New Group** from the shortcut menu. The following dialog box appears:

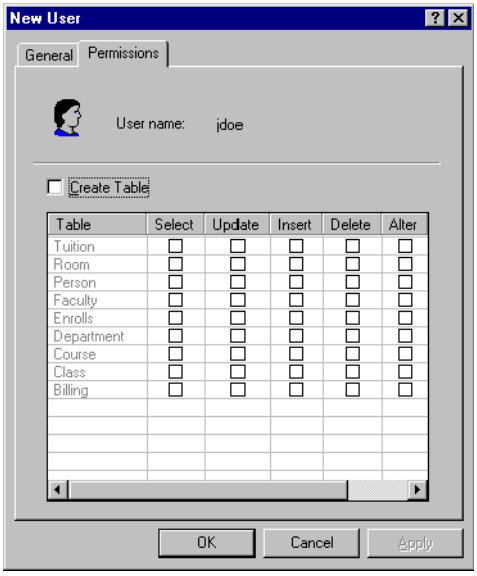
Figure 3-16 Add New Group Dialog Box



3 Type in a group name.

4 Click on the **Permissions** tab to select the specific operations that this Group is entitled to perform on the database:

Figure 3-17 Add New User / Group Permissions Dialog Box



Each check box represents a specific permission for a specific table. For example, if you want members of this group to be able to perform SELECT operations on the "Billing" table, click the box on the row labeled "Billing" under the column "Select."

5 Click **OK** when done.

You can also perform the task of creating a group and assigning permissions using the SQL statements CREATE GROUP and GRANT.

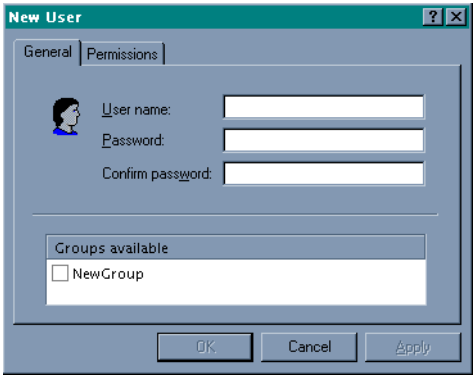
➤ To add a new user to the database

Note If you wish to use groups, you must set up the groups before creating users. You cannot add a user to a group after you have already created the user.

1 Select the Namespace node **Users** under the database in which you would like to add a new user.

2 Right-click **Users**. Select **New User** from the shortcut menu. The following dialog box appears:

Figure 3-18 Add New User Dialog Box

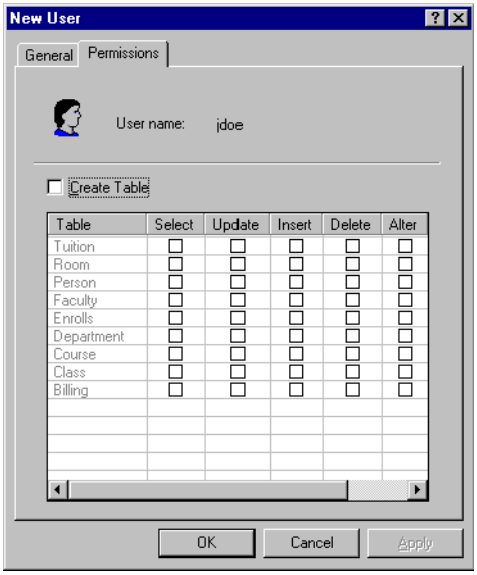


Enter the new user information in the name and password fields and select the group, if any, to which the new user belongs.

Note Group members do not have individual permissions. Every member of a group has exactly the permissions that have been defined for the group.

3 Click on the **Permissions** tab to select the specific operations that this user is entitled to perform on the database:

Figure 3-19 Add New User / Group Permissions Dialog Box



Each check box represents a specific permission for a specific table. For example, if you want members of this group to be able to perform SELECT operations on the "Billing" table, click the box on the row labeled "Billing" under the column "Select."

4 Click **OK** when done.

You can also perform the task of creating a user and assigning permissions using the SQL statement GRANT.

➤ To delete a user or group

Note To delete a group, you must first delete all users in the group. You cannot delete a group that contains users.

1 Double-click the **Users** node in PCC. Right-click on the icon representing the user or group you want to delete.

2 From the pop-up menu, click **Delete**.

You can also perform this task using the SQL statement REVOKE LOGIN FROM *user*.

➤ To add an existing user to a group

Existing users cannot be directly added to a group. Follow these steps:

1. Delete the user by following the steps provided in "To delete a user or group" on page 3-26.
2. Re-create the user by following the steps provided in "To add a new user to the database" on page 3-25.

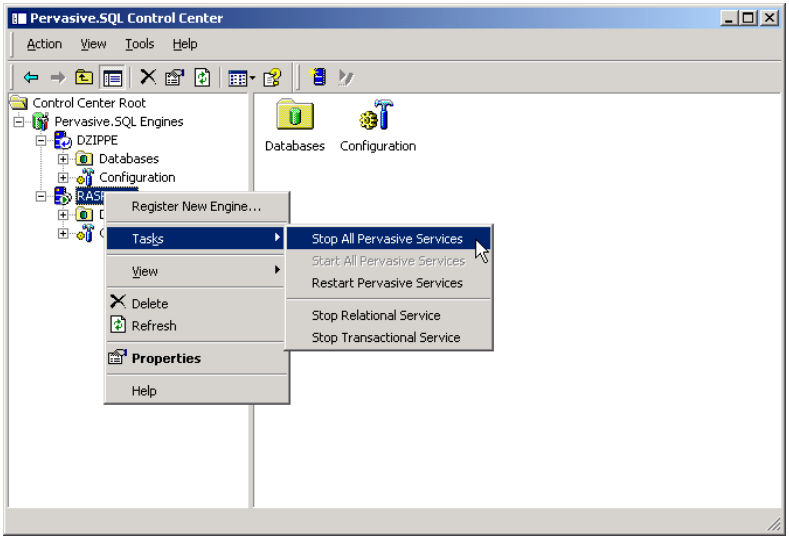
Stopping and Restarting Services on Windows Servers

PCC offers a convenient way to start and stop Pervasive.SQL servers on Windows NT/2000 machines without having to use the Windows Services control panel.

Services on remote machines may be started and stopped provided that you have database administration rights on the remote machine.

1 To start or stop a relational service, right-click on the machine in the PCC Namespace and select **Tasks** from the shortcut menu. The submenu options allow you to start or stop the transactional engine, the relational engine or both.

Figure 3-20 Stopping and Restarting Services in the Pervasive Control Center



Note You must stop the relational *and* transactional services to completely stop Pervasive.SQL. Stopping just one of the services does not stop the database engine completely.

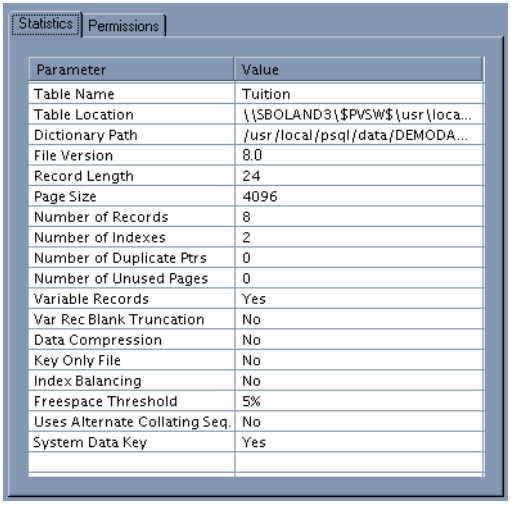
Viewing and Modifying Table Properties

In PCC, it is very easy to view the properties of a table. You can view and modify table properties as needed.

➤ To view the properties of an existing table within a database

1. Click on the **Tables** Namespace node under the database whose table you wish to view.
2. Right-click on the desired table in the table list in the Control Center's right pane.
3. Select **Properties** in the shortcut menu. The properties dialog box appears.

Figure 3-21 Table Properties Window



The Table Properties dialog box presents the following information:

Table 3-4 Existing Table Properties

Statistic Name	Description
Table Name	Shows the name of the table as it appears in the database schema.
Table Location	Shows the physical location of the data file associated with the table.
Dictionary Path	Displays where the database's DDF files are located.
File Version	Shows the earliest Btrieve engine version that can read the file format.
Record Length	Shows the length of the data file's records.
Page Size	Shows the page size (in bytes) of the data file. The page size determines the maximum number of index segments that can be defined in a table.
Number of Records	Shows the number of records currently contained in the data file.
Number of Indexes	Shows the number of indexes defined for the table.
Number of Duplicate Pointers	Shows the number of linked duplicate indexes that can be added.
Number of Unused Pages	Shows the number of pre-allocated pages available. If pre-allocation is enabled, the MicroKernel pre-allocates a specified number of pages when it creates the data file. Pre-allocation guarantees that disk space for the data file is available when the MicroKernel needs it.
Variable Records	Shows whether the data file contains variable-length records.
Variable Record Blank Truncation	Shows whether blank truncation is enabled. If it is, the MicroKernel truncates the blanks in variable-length records. Blank truncation is applicable only if the Variable Records statistic is Yes and Data Compression is set to No.
Data Compression	Shows whether data compression is enabled. If it is, the MicroKernel compresses each record it inserts into the data file.
Index Balancing	Shows whether balanced indexing is enabled.

Table 3-4 Existing Table Properties

Statistic Name	Description
Free Space Threshold	Shows a percentage (5%, 10%, 20% or 30%) if the data file has a free space threshold. The MicroKernel stores the variable-length portions of records on their own pages (called variable pages), separate from the fixed-length portions (which are stored on data pages). The MicroKernel uses the threshold to determine whether to add data to an existing variable page or to create a new one. A higher free space threshold reduces fragmentation of variable-length records across several pages but uses more disk space.
Use Alternate Collating Sequence	Shows whether the table uses an alternate collating sequence for sorting.
System Data Key	Shows whether the data file has system data keys enabled.

Occasionally, a table requires modifications to its list of columns or indexes. The Table Information Manager provides a mechanism through which such changes can be made. See the "Table Designer" on page 5-1.

Viewing and Modifying Data

When you double click a table, or right-click a table then click **Open**, PCC starts SQL Data Manager (SQL DM) to allow you to work with the data in the table.

SQL DM allows you to run Structured Query Language (SQL) statements against a Pervasive.SQL database. With SQL statements, you may retrieve, create, change, or delete data in a database provided you have the proper database permissions to perform these actions. The SQL statements that you may use with SQL DM are documented in *SQL Engine Reference*.

See "SQL Data Manager" on page 6-1 for a complete discussion of SQL DM.

Exporting/Importing Data

Pervasive.SQL provides an ODBC interface, which makes it easy to access your data from a large number of third-party programs. Sometimes data must be made available in other ways or in specific formats. To ease the migration of data to and from Pervasive.SQL databases, Pervasive Control Center (PCC) includes import and export wizards.

The Import/Export wizards in PCC support two different data formats, which can be read and written. The formats are shown in Table 3-5.

Table 3-5 Import/Export Wizards Data Formats

File Format	Description
Standard Data Format	Data is represented as standard ASCII, OEM or Unicode characters. Commas separate the columns. By default, the quote character is used to enclose columns, and carriage return/line feed is used to separate records. The column separator and record separator characters are configurable.
Unformatted MicroKernel	The wizard does not convert the data to ASCII. Binary columns remain in binary format. Each record is preceded by its length in bytes and is followed by a comma delimiter. A carriage return/ line feed terminates each record.

An empty string is designated in exported format by double quotes. A column defined as nullable and that contains a NULL is designated in exported format by two commas (two field separators) together.

Export and Import Procedures

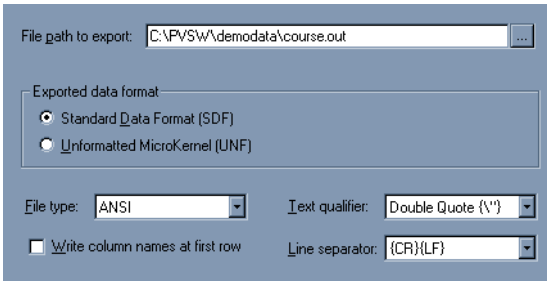
This section describes the procedures used to export or import data from or to Pervasive.SQL.

➢ To export data from a Pervasive.SQL database table

1. Select the **Tables** node in the namespace under the database from which you are exporting data.
2. Right-click on the desired table object in the table list in PCC's right pane.
3. Select **Tasks**, then **Export Data** from the shortcut menu.

4 Enter the location of the output file that you want to contain the records exported from the table.

Figure 3-22 Export Wizard - Choose a Destination File Dialog Box

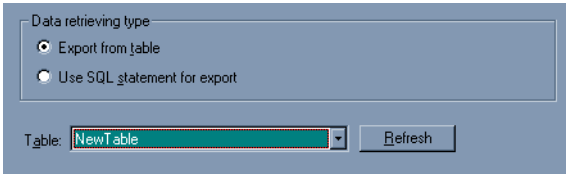


This example exports comma delimited records.

Select the box marked **Write column names at first row** if you would like the Export Wizard to create a header row at the beginning of the file that includes the name of each column from the DDF. This would be particularly useful if you are exporting to another application, such as Excel, so that the contents will be identified.

5 Click **Next**. The Export Wizard now displays a dialog box that allows records to be filtered or processed using SQL. The option chosen here is to output the table records unmodified.

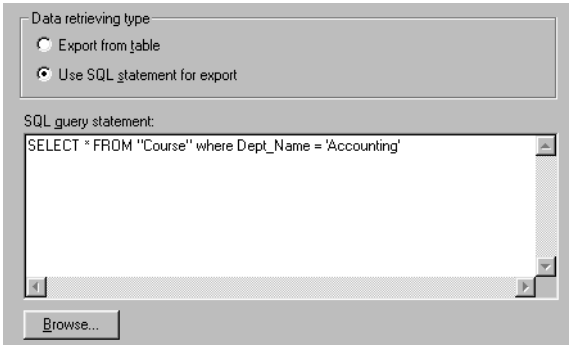
Figure 3-23 Export Wizard - Specify Table Copy or Query Dialog Box



6 Click **Use SQL statement for export** if you would like to filter the output records using SQL (Figure 3-24).

Click the **Export from table** if you want to export the entire table without specifying any filters.

Figure 3-24 Export Wizard - Use SQL Statement for Export Dialog Box



7 Click **Next**. Now that all of the information has been entered, the Export Wizard displays a dialog box listing the actions to be taken.

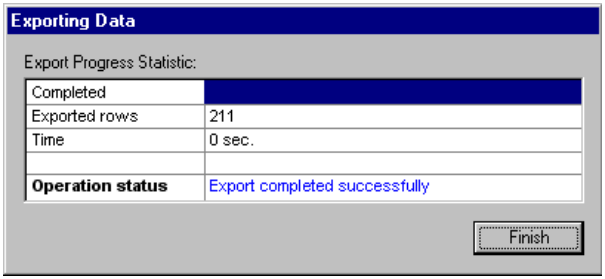
Figure 3-25 Export Wizard - Completing the Export to the File Wizard Dialog Box

Completing the Export to the File Wizard
You have successfully specified information necessary to start Export process. Review the selections below and click on EXPORT to execute.
Summary:
Source:
Server: MATTWILLSONLT
Database: DEMODATA
Export type: SQL Query:
SELECT * FROM "Course" where Dept_Name = 'Acc
Destination:
File format: SDF file
Location: c:\temp\course.out

8 Click **Export**. A dialog box provides feedback as the records are exported from the table. Once all records are processed, the **Finish** button becomes enabled. Click it to exit the wizard.

Tip If the export fails because of invalid data, you can troubleshoot the record that contains the invalid data. Click **Abort** on the error message, then **No** on the message that asks if you want to delete the partially exported file. Open the partially exported file in an editor. The last record in the file is the record containing the invalid data.

Figure 3-26 Exporting Data Status



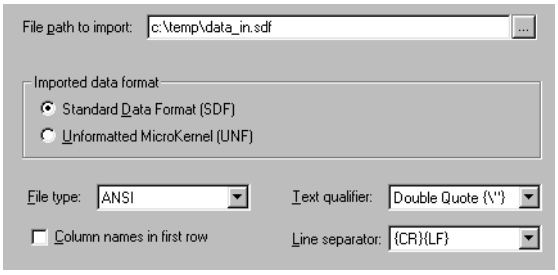
➤ To import data into an existing table

You designate an empty string by using double quotes. You designate a NULL by using two commas (two field separators) together. For example, the following table shows how you would designate a "middle initial" as an empty string and as a NULL, respectively.

Designating an Empty String	Designating a NULL
`"Jane","","Smith"`	`"Jane",,"Smith"`

1. Click on the **Tables** node in the Namespace under the database into which you are importing data.
2. Right-click on the desired table object in the table list in the Control Center's right pane.
3. Select **Tasks**, then **Import Data** in the shortcut menu.
4. Enter the location of the file that contains the data to be imported into the table. Also indicate the format of the data.

Figure 3-27 Import Wizard - Choose a Source File Dialog Box

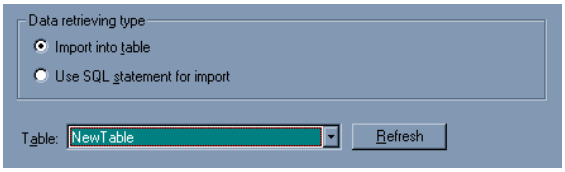


Select the box marked **Column names at first row** if you selected the option **Write column names at first row** when exported the file (Figure 3-24).

Click **Next** when done.

5 Indicate what should be done to the data as it is imported. The options are to import all of the data and insert it into the table or to specify an SQL statement, which gives you control over how the import data fields correspond to the columns in the table.

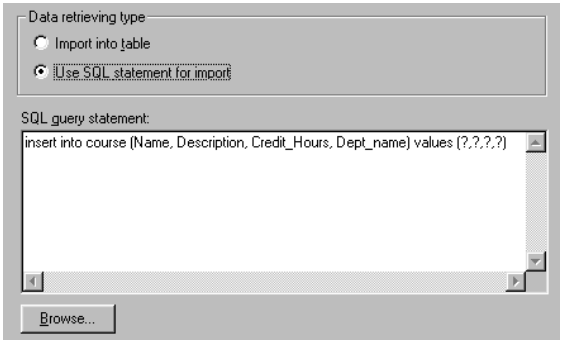
Figure 3-28 Import Wizard - Specify Table Copy Dialog Box



6 Alternatively, you can click **Use SQL statement for import** if you would like to use a SQL statement to control how the import file columns correspond to the table columns. In the example shown in Figure 3-29, the question marks (?, ?, ?, ?) represent the fields from the source file.

Select **Import into table** if you selected the option **Export from table** (Figure 3-25) to import the entire table without filters.

Figure 3-29 Import Wizard - Use SQL Statement for Import Dialog Box


Data retrieving type
Import into table
Use SQL statement for import
SQL query statement:
insert into course (Name, Description, Credit_Hours, Dept_name) values (?,?,?,?)
Browse...

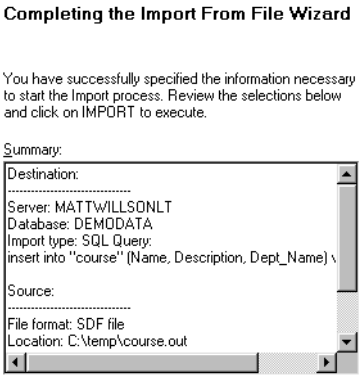
You cannot import fewer fields than exist in the table or in the import file. Both the import file and the table must have the same number of columns. The only factor you can control with the INSERT statement is how the import file columns correspond to the table columns.

The first named column receives the data from the first column in the import file, the second named column receives the data from the second column in the import file, and so on.

For example, if you want the data in the first column of the import file to go into the third column of your table, you must put the name of the third column first in the column list.

7 Now that all of the information has been entered, click **Next**. The Import Wizard displays a dialog box listing the actions to be taken.

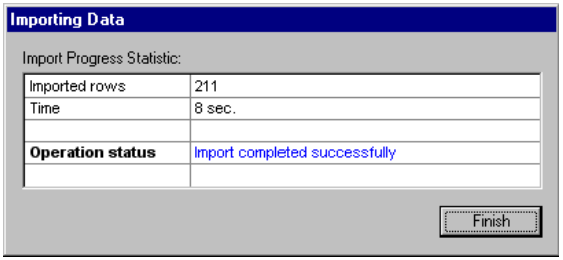
Figure 3-30 Import Wizard - Completing the Import from File Wizard Dialog Box


Completing the Import From File Wizard
You have successfully specified the information necessary to start the Import process. Review the selections below and click on IMPORT to execute.
Summary:
Destination:
Server: MATTWILLSONLT
Database: DEMODATA
Import type: SQL Query:
insert into "course" (Name, Description, Dept_Name) \
Source:
File format: SDF file
Location: C:\temp\course.out

To stop the import, click **Cancel**, or else click **Import**.

8 A dialog box provides immediate feedback as the records are imported into the table. Once all records are processed, the **Finish** button becomes enabled. Click it to exit the wizard.

Figure 3-31 Importing Data Dialog Box



➤ To import specific columns of an SDF file into a table using the import wizard

In some cases you may wish to import only some of the columns in a table into a new table. Using an example, this procedure explains how to do it.

1 First, assume we are starting with an export table that was created with the following statement:

```
CREATE TABLE Customer (FirstName char(10),
   LastName char(10),
   Telephone char(12),
   AmtOwed currency)
```

Next, assume that the following statements were executed to insert sample data:

```
INSERT INTO Customer VALUES ('Alexi', 'Pruneda',
   '512-264-9999', 1200.69)
INSERT INTO Customer VALUES ('Matte', 'Pruneda',
   '512-264-9999', 9999.69)
```

2 Now assume the import table was created with the following statement:

```
CREATE TABLE phonelist (FirstName char(10),
   LastName char(10),
   Telephone char(12))
```

Notice that this table does not have all the columns of the original one.

3 Using the following query in the Export Wizard, export the data from the original table into an export file named `cust.sdf`:

```
SELECT * FROM Customer
```

Note that the export file contains four columns:

```
"Alexi","Pruneda","512-264-9999","1200.69"
"Matt","Pruneda","512-264-9999","9999.69"
```

4 Using the Import Wizard, import the data from the SDF file into the new table.

a. To start the Import Wizard from PCC, select the **Tools** menu and choose **Wizards ▸ Import Data**.

b. In the Import Wizard, in the box labelled **File path to import**, select cust.sdf (the file that you used to export the values). Then click **Next**.

c. Choose **Use SQL statement for import**.

d. In the query window, type this query:

```
INSERT INTO phonelist(FirstName,
                      LastName,
                      Telephone)
   VALUES (?, ?, ?)
```

The named fields must be in the same order as the fields listed in the `CUST.SDF` file. The named fields do not need to be in the same order as they exist in your destination table. You can move the column names in the INSERT statement around as necessary to correspond properly with the fields in the SDF file. Click **Next**.

e. You should get immediate feedback that the records were imported. Note that the fourth data column in the SDF file is ignored.

Notes

The import process can exclude any number of fields at the end of each record, but it cannot exclude any fields at the beginning or middle of a record. As in the example above, the SDF file contained 4 fields, but we imported only the first 3 by including only 3 question marks in the VALUES clause. You may not exclude beginning or middle values from the SDF file, as in `VALUES ( ,?,?,?)` or `VALUES(?, ,?,?)`. These syntax forms generate error messages. If you need to achieve this effect, then you should use a SELECT statement during your export process to generate an SDF file that contains only the columns you want to import into your other table.

The process is simpler if you are exporting and importing the same number of fields to and from the SDF (standard data format) files.

Then the SQL queries are not necessary, and you can choose to export and import the full table.

If the export and import tables are part of the same database, importing can also be done using SQL queries. The following example would achieve the same results as the steps listed above:

```
INSERT INTO phonelist (FirstName, LastName, Telephone)
SELECT FirstName, LastName, Telephone FROM Customer
```

Checking Consistency and Referential Integrity

Checking the consistency and the integrity of a file and displaying the constraints on a file is a simple process using the Check Database wizard in PCC. Using this wizard, you can perform a consistency test, a referential integrity (RI) test or an index consistency test.

A consistency test checks logical and physical consistency of the selected database, including table and indexes; a referential integrity test checks referential integrity constraints applied to the selected database.

To perform the tests, you must be logged into the database using its data source name.

If you are performing the RI test, your database must be named, stored on the same server as the SRDE, and have referential constraints defined (whether or not RI is enabled).

Listing Referential Constraints

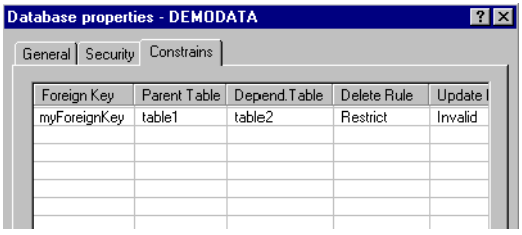
➢ **To produce a report that lists all foreign key definitions and related information for any database**

1 Right-click on the database in the PCC Namespace whose referential constraints are to be viewed.

2 Select **Properties** from the shortcut menu.

3 Click on the tab marked **Constraints.** The following dialog box appears:

Figure 3-32 Database Properties



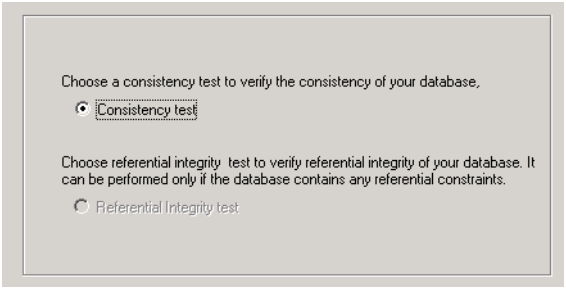
Click **OK** to close the window.

Checking Consistency

➢ To check whether the table definitions in your DDFs match the file structure of the data files

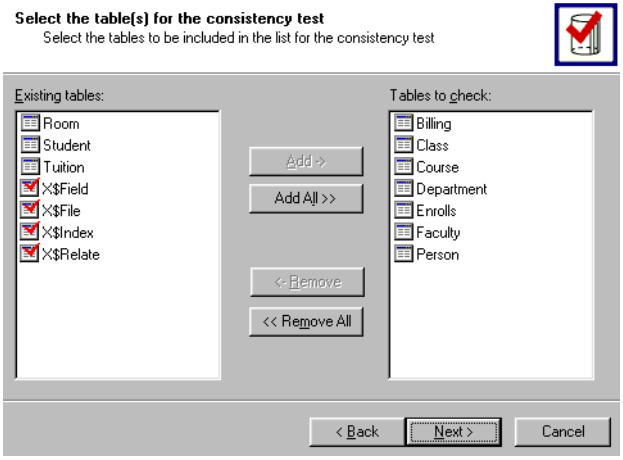
1. Right-click on the database in the PCC Namespace whose referential integrity is to be checked.
2. Select **Tasks**, then **Check Databases** from the shortcut menu. The following dialog box appears:

Figure 3-33 Check Database Dialog Box


Choose a consistency test to verify the consistency of your database,
Consistency test
Choose referential integrity test to verify referential integrity of your database. It can be performed only if the database contains any referential constraints.
Referential Integrity test

3. Click **Consistency test**, then click **Next**. The select table dialog box allows you to specify which tables of the database are to be checked:

Figure 3-34 Select Tables for the Referential Integrity Test Dialog Box



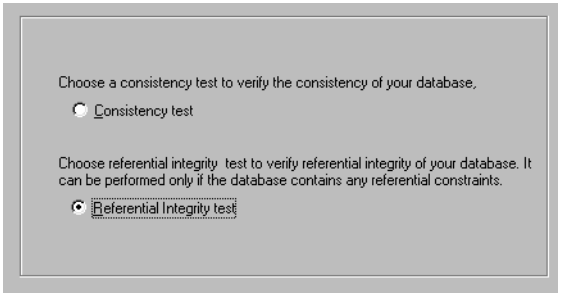
Select the files you want to check, click **Next**, and follow the remaining prompts.

Checking Referential Integrity

➢ To check for orphan rows in a database or verify the consistency of a data file's referential integrity constraints

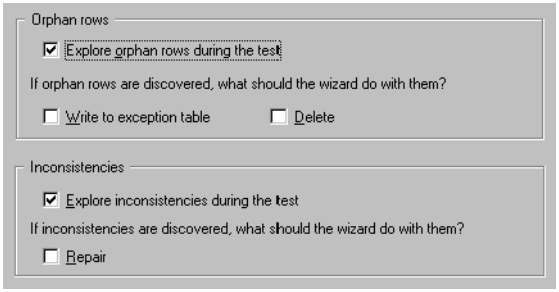
1. Right-click on the database in the PCC Namespace whose referential integrity is to be checked.
2. Select **Tasks**, then **Check Databases** from the shortcut menu. The following dialog box appears:

Figure 3-35 Check Database Dialog Box


Choose a consistency test to verify the consistency of your database,
Consistency test
Choose referential integrity test to verify referential integrity of your database. It can be performed only if the database contains any referential constraints.
Referential Integrity test

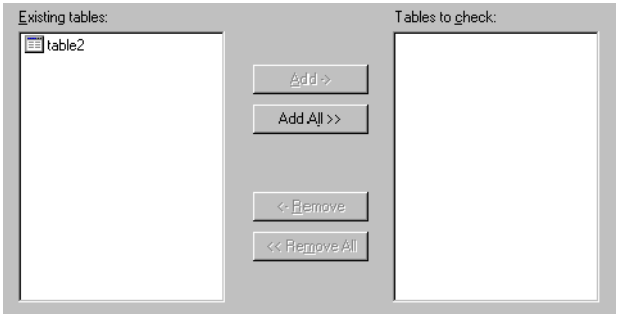
3. Select **Referential Integrity test** and click **Next**.
4. The **Details** dialog box provides options to specify what actions should be taken when inconsistencies or orphan rows are found. Select any desired options and click **Next**.

Figure 3-36 Details of the Referential Integrity Test Dialog Box


Orphan rows
Explore orphan rows during the test
If orphan rows are discovered, what should the wizard do with them?
Write to exception table
Delete
Inconsistencies
Explore inconsistencies during the test
If inconsistencies are discovered, what should the wizard do with them?
Repair

5. The select table dialog box allows you to specify which tables of the database are to be checked:

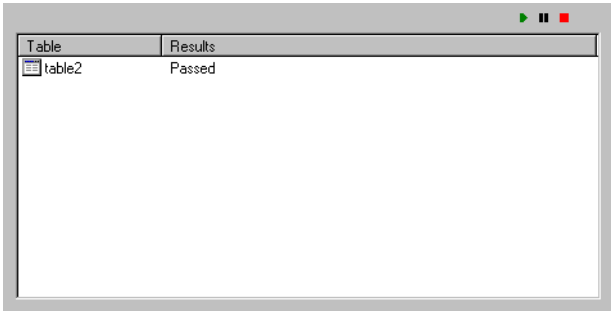
Figure 3-37 Select Tables for the Referential Integrity Test Dialog Box



Make your selections, then click **Next**.

6 The resulting dialog box shows the results of the referential integrity check on the tables as the wizard checks each table selected:

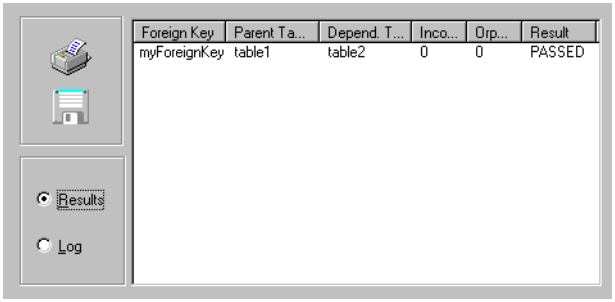
Figure 3-38 Check Database Results Dialog Box



When the wizard has finished checking all tables, click **Next**.

7 The Check Database Wizard's result dialog box shows a complete list of foreign keys and the results of each test. You can save the test results to disk or print them out at this time:

Figure 3-39 Referential Integrity Test Results Dialog Box



8 Follow the wizard prompts until the you receive confirmation that the tests are complete.

To repair inconsistencies, the Check Database wizard updates the information stored in the individual data files to match that in the data dictionary.

chapter

4

License Administrator

Working with License Keys and User Counts

This chapter covers the following topics:

- "License Administrator Concepts" on page 4-2
- "License Administrator GUI Visual Reference" on page 4-5
- "License Administrator Tasks" on page 4-9

License Administrator Concepts

A Pervasive.SQL license key allows a fixed number of computers (called a user count) to access a Pervasive.SQL engine concurrently. A license key is human-readable text (a string of letters and numbers). You can cut and paste the text, for example, to speed the entry of a key during installation of a Pervasive.SQL product.

The License Administrator utility allows you to apply license keys, remove licenses, and view license information. The utility includes a graphical user interface (GUI) and a command line interface (CLI).

Graphical User Interface

The GUI runs only on Windows 32-bit platforms and allows you to apply license keys, remove licenses, and view license information. You can start the GUI as a stand-alone application or from within Pervasive Control Center (PCC). You may administer licenses for local or remote database engines with the GUI.

See "License Administrator GUI Visual Reference" on page 4-5 for an explanation of the objects on the GUI. See "GUI Tasks" on page 4-10 for the tasks specific to the graphical user interface.

Command Line Interface

The CLI runs on all platforms supported by Pervasive.SQL. The CLI is functionally equivalent to the GUI with two exceptions. You may administer licenses only for local database engines with the CLI. In addition, the CLI requires that you know a license key (the string of letters and numbers) to remove the license.

The GUI, by contrast, lists all applied licenses and allows you to select a license to remove. The GUI does not list the license *keys*, just the information *about* the applied licenses, such as the type of license, user count, expiration date, and so forth.

For security, the CLI does not display the license keys themselves. (You may obtain the license key from the installation media.) You can, however, just as with the GUI, display information about the licenses applied to the computer (type of license, user count, and so forth).

You may administer licenses only for the computer on which the CLI is running. The CLI cannot administer remote licenses.

The CLI uses the same syntax on all platforms:

utility_name -option parameters

The utility name is one of the following:

- clilcadm.exe (Windows)
- clilcadm.nlm (NetWare)
- clilcadm (Linux)

Table 4-1 describes the options.

Table 4-1 License Administrator Command Line Options and Parameters

Options and Parameters	Function
-<interpret | i>	Displays information about all of the applied licenses, such as the type of license, user count, expiration date, and so forth. This option does not display the license keys themselves (the string of letters and numbers).
-<interpret | i> *license_key*	Displays information about a specific license, such as the type of license, user count, expiration date, and so forth.
-<apply | a> *license_key*	Applies the specified license key to the computer on which the CLI utility is running.
-<remove | r> *license_key*	Deletes the specified license from the computer on which the CLI utility is running.
-<help | h>	Displays information about the options and parameters.
no option or wrong option	Displays information about the options and parameters.

See “CLI Tasks” on page 4-9 for the tasks pertaining to the CLI.

User Count

Each license specifies a user count. A user count allows the specified number of computers to connect to the Pervasive database engine concurrently. Users are counted by network address. The IP address is used for TCP/IP; the IPX address is used for SPX/IPX.

Each workstation that accesses Pervasive.SQL as a client counts as one user. Multiple applications on a single client computer are counted as one user, not separate users. Each Terminal Server session also counts as one user.

Collectively, all applications that access the database engine, use the same network protocol and address, and run on the same machine as the database engine count as one user. If one application uses TCP/IP and another application uses SPX/IPX, two licenses are counted if both applications run on the same machine. Similarly, if a machine contains multiple network cards, a user would be counted for each unique network address being used.

Obtaining a User Count

A user count is obtained in the form of a software license key issued by Pervasive Software, or by your application vendor if the Pervasive.SQL database engine is embedded in an application.

The Pervasive.SQL Server engine includes a temporary trial license that allocates a fixed user count. The trial license expires within a specified period and cannot be deleted. The Pervasive.SQL Workgroup engine includes a license set to five concurrent users. The license for the Workgroup engine cannot be deleted.

Please contact Pervasive Software or your application vendor to purchase an additional user count.

Applying a User Count

You increase the user count by applying a license key. A user count is applied immediately when you use the GUI or the CLI.

Licenses for Prior Versions of Pervasive.SQL

Pervasive.SQL 2000i and prior releases use a different type of license. License Administrator automatically detects the database engine version to which you want to apply a license. For Pervasive.SQL 2000i and prior releases, you must use the legacy utility, User Count Administrator, to work with the older license type. User Count Administrator is installed as part of the prior release's database engine.

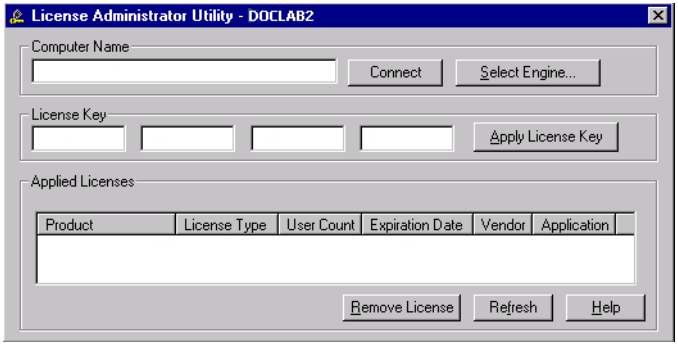
Refer to the Pervasive.SQL documentation of the prior release for instructions on how to use User Count Administrator. Pervasive.SQL documentation is located on the Web at http://www.pervasive.com/support/technical/online_manuals.asp.

License Administrator GUI Visual Reference

This section describes the objects on the License Administrator graphical user interface (GUI).

GUI Dialog

The following image shows the GUI. The table below the image describes the GUI objects.



GUI Object	Description	Related Information
Title Bar	Shows the name of the utility and the name of the computer that License Administrator is accessing. By default, the computer name is the local computer to which you are currently logged in.	
Computer Name	By default, shows the name of the computer that the License Administrator is accessing. This object is also an entry field into which you may type the name of a computer. If you have typed in a computer name but not clicked **Connect** (or pressed **Enter**), the field could contain a computer name that differs from the one in the title bar. The title bar always shows the computer that License Administrator is accessing. The **Computer Name** field may also be populated if you click **Select Engine** and choose a computer.	"To Select a Computer for License Administration" on page 4-11
Connect	Establishes communication between License Administrator and the computer listed in the **Computer Name** field.	"To Select a Computer for License Administration" on page 4-11
Select Engine	Allows you to choose a computer on which to administer licenses.	"To Select a Computer for License Administration" on page 4-11
License Key	Accepts a license key (a combination of letters and numbers) typed in or pasted.	"License Administrator Concepts" on page 4-2 "To Apply a License Key" on page 4-13

GUI Object	Description	Related Information
Apply License Key	Applies the license key in the **License Key** field to the specified database engine. The title bar shows the computer on which the database engine is running. This button is enabled if the license key in the **License Key** field is valid and a database engine is running on the specified computer.	"To Apply a License Key" on page 4-13
Applied Licenses	Lists the applied license information for the specified computer: ◆ Product – The name of the Pervasive.SQL product, such as Server or Workgroup. ◆ License Type – The type of license, such as permanent or temporary. ◆ User Count – The user count for the specific license. ◆ Expiration Date – The date on which a license expires, if applicable, or "n/a" if not applicable. ◆ Vendor – A unique number identifying the vendor software that installed the license. ◆ Application – A unique number identifying the application to which the license applies.	"User Count" on page 4-3 "To Display Applied Licenses" on page 4-14
Remove License	Removes the selected license(s) from the specified computer. This button is enabled when a Product name is selected in the **Applied Licenses** list.	"To Remove a License" on page 4-14

GUI Object	Description	Related Information
Refresh	Re-displays the information for Applied Licenses and removes any license key information from the License Key field. The Computer Name field is cleared then displays the machine name to which License Administrator is currently connected.	“To Refresh the List of Applied Licenses” on page 4-15
Help	Displays the online help for License Administrator.	“To Display Help” on page 4-15

License Administrator Tasks

This section explains the tasks that you can perform with License Administrator.

GUI Tasks

- "To Start License Administrator from Pervasive Control Center (PCC)" on page 4-10
- "To Start License Administrator as a Stand-alone Application" on page 4-11
- "To Select a Computer for License Administration" on page 4-11
- "To Apply a License Key" on page 4-13
- "To Remove a License" on page 4-14
- "To Display Applied Licenses" on page 4-14
- "To Determine a Total User Count" on page 4-15
- "To Refresh the List of Applied Licenses" on page 4-15
- "To Display Help" on page 4-17

CLI Tasks

- "To Display All Applied Licenses" on page 4-16
- "To Display a Specific Applied License" on page 4-16
- "To Apply a License Key" on page 4-16
- "To Remove a License" on page 4-16
- "To Display Help" on page 4-17

GUI Tasks

➢ To Start License Administrator from Pervasive Control Center (PCC)

1 In PCC namespace tree, click on a computer name or on the word "Databases" for a specific computer. Choose a computer for which you want to administer licenses.

2 Click **Tools ▸ License Administrator.**

License Administrator accesses the selected computer to determine the type of license required. One of three actions then occurs:

- If the selected computer is running a Pervasive.SQL release issued *after* Pervasive.SQL 2000i and if you are logged in to the computer, License Administrator connects to the database engine. The computer name appears in the License Administrator title bar and in the **Computer Name** field.
- If the selected computer is running a Pervasive.SQL release issued *after* Pervasive.SQL 2000i and if you are *not* logged in to the computer, a dialog appears from which you log in to the operating system. Note that you are logging in to the operating system and not to the database engine.

  On the login dialog, type the name of an operating system user for **User Name**. The user must have administrative privileges on the operating system where the database engine is running. For **Password**, type the appropriate password for the user name you specified. Click **Login**. The computer name appears in the License Administrator title bar and in the **Computer Name** field.
- If the selected computer is running Pervasive.SQL 2000i or a prior release, a message displays informing you that License Administrator does not support the prior release. The message instructs you to use the legacy utility, User Count Administrator, to work with the older license type. User Count Administrator is installed as part of the prior release's database engine.

If the computer is not the one you want, select a different computer as described in "To Select a Computer for License Administration" on page 4-11.

Note Pervasive.SQL 2000i and prior releases use a different type of license. License Administrator automatically detects the license type on the machine being accessed. For Pervasive.SQL 2000i and prior releases, you must use the legacy utility, User Count Administrator, to work with the older license type. User Count Administrator is installed as part of the prior release's database engine.

Refer to the Pervasive.SQL documentation of the prior release for information on how to use User Count Administrator. Pervasive.SQL documentation is located on the Web at http://www.pervasive.com/support/technical/online_manuals.asp.

➤ To Start License Administrator as a Stand-alone Application

1 From the desktop, click **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Other Utilities ▸ License Administrator.**

This step assumes that License Administrator was installed as part of a Pervasive.SQL default installation. A vendor application that uses an embedded database engine may require that you start License Administrator with different menu commands. (The GUI executable is named guilcadm.exe.)

The utility connects to the local computer on which the database engine is running.

If the computer is not the one you want, select a different computer as described in "To Select a Computer for License Administration" on page 4-11.

➤ To Select a Computer for License Administration

The database engine must be running on the computer you specify in order for License Administrator to connect to the engine.

1 In the **Computer Name** field, type the name of the computer for which you want to administer database licenses.

As an alternative to typing a computer name, you may click **Select Engine** to display a list of network computers. Expand the list of computers as needed (click the plus sign). Click on the computer for which you want to administer database licenses, then click **OK**.

2 Click **Connect** (or press **Enter**).

License Administrator accesses the specified computer to determine the type of license required. One of three actions then occurs:

- If the selected computer is running a Pervasive.SQL release issued *after* Pervasive.SQL 2000i and if you are logged in to the computer, License Administrator connects to the database engine. The computer name appears in the License Administrator title bar and in the **Computer Name** field.
- If the selected computer is running a Pervasive.SQL release issued *after* Pervasive.SQL 2000i and if you are not logged in to the computer, a dialog appears from which you log in to the operating system. Note that you are logging in to the operating system and not to the database engine.

  On the login dialog, type the name of an operating system user for **User Name**. The user must have administrative privileges for the database engine. For **Password**, type the appropriate password for the user name you specified. Click **Login**. The computer name appears in the License Administrator title bar and in the **Computer Name** field.
- If the selected computer is running Pervasive.SQL 2000i or a prior release, a message displays informing you that License Administrator does not support the prior release. The message instructs you to use the legacy utility, User Count Administrator, to work with the older license type. User Count Administrator is installed as part of the prior release's database engine.

Note Pervasive.SQL 2000i and prior releases use a different type of license. License Administrator automatically detects the license type on the machine being accessed. For Pervasive.SQL 2000i and prior releases, License Administrator gives you the option of starting the legacy utility, User Count Administrator, to work with the older license type.

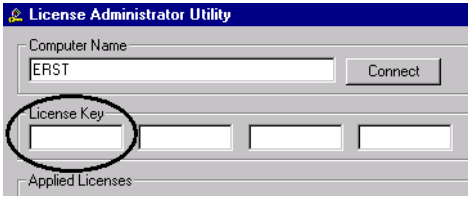
Refer to the Pervasive.SQL documentation of the prior version for how to use User Count Administrator. Pervasive.SQL documentation is located on the Web at http://www.pervasive.com/support/technical/online_manuals.asp.

➤ To Apply a License Key

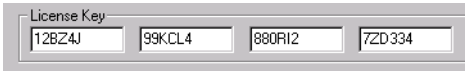
1. Ensure that the computer name in the title bar is the computer for which you want to add a database license. If not, select a different computer as described in "To Select a Computer for License Administration" on page 4-11.
2. Type, or paste, the license key into the **License Key** field.

   Lowercase letters are automatically converted to uppercase.

   Paste the key into the first (the left) field.



   The key is distributed across the remaining fields.



3. Press **Enter** or click **Apply License Key**.

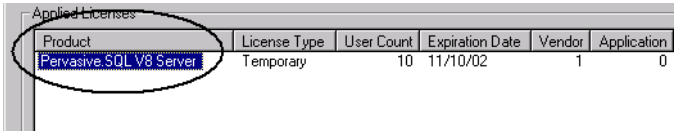
   The License Key field is cleared if the license key is successfully applied.
4. Verify the new license information that appears in the **Applied Licenses** list. (If necessary, click **Refresh** to refresh the list.)

The license is now active. You do not have to restart the database engine.

➤ To Remove a License

Note Temporary licenses cannot be removed. They expire at the end of their activation period. The trial license included with the Pervasive.SQL Server engine cannot be deleted because it is a temporary license. The 5-user license included with the Pervasive.SQL Workgroup engine cannot be deleted.

1. Ensure that the computer name in the title bar is the computer from which you want to delete a database license. If not, select a different computer as described in "To Select a Computer for License Administration" on page 4-11.
2. Click on a **Product** name in the **Applied Licenses** list.



   You may select multiple licenses by holding down the **Shift** or **Control** keys and clicking on the desired product names.
3. Click **Remove License**.

➤ To Display Applied Licenses

1. Ensure that the computer name in the title bar is the computer for which you want to administer database licenses. If not, select a different computer as described in "To Select a Computer for License Administration" on page 4-11.
2. View the information for the applied licenses in the **Applied Licenses** list.

Note You must first apply a license to display information about it, such as user count, license type, expiration date and so forth. Once applied, if the license key is not what you want, you may remove it provided the license is not a temporary license. Temporary licenses expire on their own and cannot be removed. The permanent license

included with the Pervasive.SQL Workgroup engine cannot be deleted.

See "To Apply a License Key" on page 4-13 and "To Remove a License" on page 4-14.

➤ To Determine a Total User Count

1. Ensure that the computer name in the title bar is the computer for which you want determine a database user count. If not, select a different computer as described in "To Select a Computer for License Administration" on page 4-11.
2. For a particular product, total the user count values that appear in the **Applied Licenses** list, excluding any values for expired temporary licenses. For example, total all values for the Pervasive.SQL Server product that are permanent licenses or temporary licenses that have not expired.

   The sum is the total number of users who can concurrently connect to the database engine on the computer.

➤ To Refresh the List of Applied Licenses

1. Ensure that the computer name in the title bar is the computer you want. If not, select a different computer as described in "To Select a Computer for License Administration" on page 4-11.
2. Click **Refresh**.

   The information for Applied Licenses is re-displayed, and any license key information is removed from the License Key field.

   The Computer Name field is cleared then displays the machine name to which License Administrator is currently connected. For example, suppose License Administrator is connected to PVSW1 and you attempt to connect to PVSW2, which does not have a database engine running. License Administrator displays a message that it could not connect, and PVSW2 is displayed in the Computer Name field. After you click Refresh, PVSW1 is displayed in the field.

➤ To Display Help

1. Click **Help**.
2. Click the desired area:

- License Administrator Concepts — to understand the basics of License Administrator
- License Administrator GUI Visual Reference — to identify and understand the purpose of the objects on the GUI
- License Administrator Tasks — to perform activities with License Administrator.

CLI Tasks

➤ To Display All Applied Licenses

1 Enter the following command:

```
clilcadm -<interpret | i>
```

The license key is located on the Pervasive.SQL product installation media. NetWare 4.x requires the word "load" (load clilcadm).

➤ To Display a Specific Applied License

1 Enter the following command:

```
clilcadm -<interpret | i> license_key
```

NetWare 4.x requires the word "load" (load clilcadm).

➤ To Apply a License Key

1 Enter the following command:

```
clilcadm -<apply | a> license_key
```

The license key is located on the Pervasive.SQL installation media. NetWare 4.x requires the word "load" (load clilcadm).

➤ To Remove a License

1 Enter the following command:

```
clilcadm -<remove | r> license_key
```

The license key is located on the Pervasive.SQL installation media. NetWare 4.x requires the word "load" (load clilcadm).

➤ To Display Help

1 Enter either of the following commands:

```
clilcadm -<help | h>
clilcadm
```

NetWare 4.x requires the word "load" (load clilcadm).

chapter

5

Table Designer

A Tour of Pervasive Table Designer

The topics in this chapter include:

- "Table Designer Concepts" on page 5-2
- "Table Designer GUI Visual Reference" on page 5-8
- "Table Designer Tasks" on page 5-28

Table Designer Concepts

This section contains the following topics:

- "Overview" on page 5-2
- "Tabbed Dialogs" on page 5-3
- "Modes" on page 5-5
- "Data Types" on page 5-6
- "Null Values" on page 5-6

Overview

The Pervasive.SQL Table Designer allows you to add, delete, or change the characteristics of columns within a table. You may apply these changes only to the data definition files, files.ddf, field.ddf, and index.ddf, or also to the actual data in the data file (such as .mkd files). In addition, Table Designer allows you create a database or a table.

To create or modify a database with Table Designer, Btrieve database security must be turned off. In addition, you must have full administrator rights on the machine on which the database engine is running *even if* you are a member of the Pervasive_Admin group. See "Database Security" on page 1-8 in *Advanced Operations Guide* for more about Btrieve database security and operating system access rights, and "Granting Administrative Rights for the Database Engine" on page 2-10.

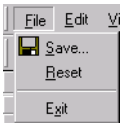
Caution Backup all your data definition files (DDFs) and data files before you perform functions through Table Designer. This tool gives you the ability to modify your database table definitions and data. If you inadvertently set the options incorrectly or enter incorrect data, you could change your files in an irreversible manner. Full recovery is possible if you have performed a backup.

Table Designer includes a graphical user interface (GUI) that runs only on Windows 32-bit platforms. You may start the GUI as a stand-alone application or from within Pervasive Control Center (PCC). See "Table Designer GUI Visual Reference" on page 5-8 for an explanation of the objects on the GUI.

When running stand-alone, the tool can access databases and tables only on the machine from which you started Table Designer. The **File**

menu commands differ somewhat depending on whether you start Table Designer stand-alone or from PCC.

Table 5-1 Comparison of File Menu Commands

Table Designer Started Stand-alone	Table Designer Started from PCC
	

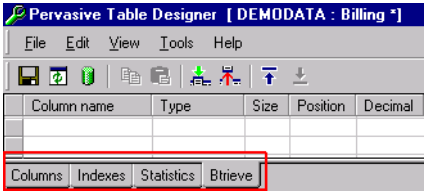
The commands that allow you to work with databases and tables are absent when you start Table Designer from PCC. PCC already provides this functionality through its own commands and wizards.

Tabbed Dialogs

Table Designer contains the following tabbed dialogs that you use as work areas:

- Columns
- Indexes
- Statistics
- Btrieve

Figure 5-1 Tabbed Dialogs in Table Designer



You select a tabbed dialog by clicking on its tab.

Columns Tab

The Columns tab lets you add and modify columns when Table Designer is in linked mode. When Table Designer is in unlinked mode, the Columns tab is read-only. See "Modes" on page 5-5.

The information on the Columns tab directly corresponds to the information on the Btrieve tab. This correspondence lets you see the Btrieve equivalents on the Btrieve tab. The Btrieve tab appears when Table Designer is in unlinked mode.

See "Columns Tab" on page 5-18 for a description of the areas on the Columns tab. See "Columns Tasks" on page 5-29 for the tasks that you perform on the tab.

Indexes Tab

The Indexes tab lets you add and modify indexes and index segments. When Table Designer is in unlinked mode, the index changes affect only the data definitional files. See "Modes" on page 5-5.

See "Indexes Tab" on page 5-20 for a description of the areas on the tab. See "Indexes Tasks" on page 5-29 for the tasks that you perform on the tab.

Statistics Tab

The Statistics tab shows you information about the current table. You may modify the location of a table's data file on this tab. All other information on this tab is read-only.

See "Statistics Tab" on page 5-21 for a description of the areas on the tab and "Statistics Tasks" on page 5-30 for the tasks that you perform on the tab.

Btrieve Tab

The Btrieve tab displays data in a Btrieve file and lets you modify the data definition files. The Btrieve tab appears only if Table Designer is in unlinked mode. See "Unlinked Mode" on page 5-5.

The information on the Btrieve tab directly corresponds to the information on the Columns tab. This correspondence lets you see the SQL equivalents on the Columns tab. Note that when Table

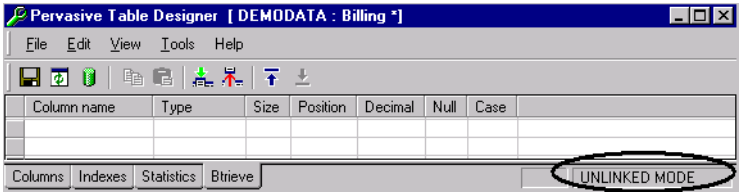
Designer is in unlinked mode—that is, when the Btrieve tab is available—the Columns tab is read-only.

See "Btrieve Tab" on page 5-25 for a description of the areas on the tab. See "Btrieve Tasks" on page 5-30 for the tasks that you perform on the tab.

Modes

Table Designer functions in one of two modes: linked or unlinked. The lower right corner of the main window indicates your current mode. Also note that all four tabbed dialogs appear only in the unlinked mode.

Figure 5-2 Mode Indicator



Linked Mode

When Table Designer is in linked mode, the changes you make are reflected in both the table definitions and the corresponding data files. That is, the changes are *linked* between the table definitions and the data. The changes affect files.ddf, field.ddf, and index.ddf, and data files such as .mkd files.

Linked mode is the default mode. In this mode, only the Columns tab, Indexes tab, and Statistics tab appear. Use this mode to design your table characteristics in terms of SQL access.

Unlinked Mode

When Table Designer is in unlinked mode, the changes you make are reflected only in the table definitions. The data files are not modified in any way. The changes are *not* linked between the table definitions and the data. The changes affect only data definition files such as file.ddf, field.ddf, and index.ddf.

In this mode, all four of the tabbed dialogs appear, but the Columns tab is read-only. Use this mode to modify table definitions to match existing data files. This mode also allows you to view the mappings

between Btrieve data types and SQL data types. See "Data Types" on page 5-6.

Data Types

Refer to "Pervasive.SQL Supported Data Types" on page A-2 in *SQL Engine Reference* for a list of the data types supported by the database engine. That section lists the Pervasive.SQL data types for the transactional and relational interfaces and the equivalent ODBC data types. You may use any data types listed in "Pervasive.SQL Supported Data Types" *and* that appear in the "Type" selection list on the Columns tab or the Btrieve tab.

The following data types are supported by the relational interface as *read-only* data types. They cannot be created in an SQL statement.

- LVAR
- NOTE
- WSTRING
- WZSTRING

Null Values

The ability to modify the null attribute of a column is subject to the following restrictions:

- The target column cannot have a PRIMARY/FOREIGN KEY constraint defined on it.
- If converting the old type to the new type causes an overflow (arithmetic or size), the ALTER TABLE operation is aborted.
- If a nullable column contains NULL values, the column cannot be changed to a non-nullable column.

If you must change the data type of a key column, you can do so by deleting the index key, changing the data type, and re-adding the key. Keep in mind that you must ensure that all associated index key columns in the database remain synchronized.

For example, if you have a primary index key in table T1 that is referenced by foreign keys in tables T2 and T3, you must first delete the foreign keys. Then you delete the primary key and change all three columns to the same data type. Finally, you must re-add the primary key and then the foreign keys.

For additional information on nulls, see the following:

- "Null Conversion" on page 15-2 in *Advanced Operations Guide*
- "INSERT" on page 3-85 in *SQL Engine Reference*

- Null Value in *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

PVS File

When you create a database with Table Designer, a file with a `pvs` file extension is created in the same directory location as you specified for the database. (See steps 2 and 3 in "To create a database" on page 5-34.) The pvs file contains the same name that you specified for the database.

The pvs file is a shortcut file that provides a convenient way to open Table Designer. Double-click the file name from the operating system file explorer to open Table Designer. The list of databases displays and the database associated with the pvs file is selected. Click **OK** to open the database.

For example, if you create a database named `DBAcct` in `C:\MyDatabases`, Table Designer also creates `DBAcct.pvs` in `C:\MyDatabases`. If you double-click `DBAcct.pvs` (or right-click the file then click **Open**), Table Designer opens to the list of databases and the name "DBAcct" is selected in the list.

You may delete the pvs file if you choose. It is not required by Table Designer but provided solely as a convenience for you. To re-create a pvs file, create an empty file with an ASCII editor and name the file the same as the database name but with a pvs file extension.

Table Designer GUI Visual Reference

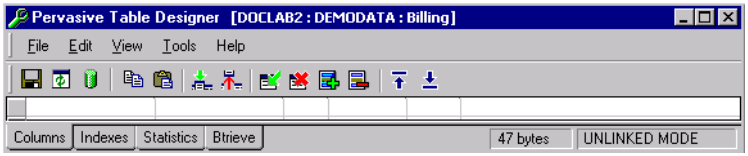
This section describes the objects on the Table Designer graphical user interface (GUI).

- Main Window
- Columns Tab
- Indexes Tab
- Statistics Tab
- Btrieve Tab

Main Window

The following image shows the main window of the GUI. (The image shows all icons that may appear on the window; not all icons appear for each tabbed work area.) The table below the image describes the GUI objects.

Figure 5-3 Table Designer Main Window



GUI Object	Description	Related Information
Title Bar	Shows the following information: ◆ name of the utility ◆ name of the computer on which the database resides (if Table Designer started from PCC) ◆ name of the open database ◆ name of the open table (For example, in Figure 5-3, table Billing in database DEMODATA on machine DOCLAB2)	
File menu	Provides the following commands: ◆ Save ◆ Reset ◆ Open Table ◆ Open Database ◆ Create Table ◆ Create Database ◆ Delete Table ◆ Delete Database ◆ Exit	
Save command or icon	Saves the Table Designer information to catalog files.	“To save changes” on page 5-37

GUI Object	Description	Related Information
Reset	Restores the information from the last saved state. All changes that have not been saved are discarded.	"To discard all changes (Reset)" on page 5-37
Open Table	Opens a table. This command appears only if Table Designer is running as a stand-alone application.	"To start Table Designer as a stand-alone application" on page 5-30 "To open a table" on page 5-35
Open Database	Opens a database. This command appears only if Table Designer is running as a stand-alone application.	"To start Table Designer as a stand-alone application" on page 5-30 "To open a database" on page 5-33
Create Table	Creates a new table in your database and opens that table for you. This command appears only if Table Designer is running as a stand-alone application.	"To start Table Designer as a stand-alone application" on page 5-30 "To create a table" on page 5-36
Create Database	Creates a new database, which then becomes the one open in Table Designer. This command appears only if Table Designer is running as a stand-alone application.	"To start Table Designer as a stand-alone application" on page 5-30 "To create a database" on page 5-34
Delete Table	Deletes a table from physical storage. This command appears only if Table Designer is running as a stand-alone application.	"To start Table Designer as a stand-alone application" on page 5-30 "To delete a table" on page 5-36

GUI Object	Description	Related Information
Delete Database	Deletes a database from physical storage. This command appears only if Table Designer is running as a stand-alone application.	"To start Table Designer as a stand-alone application" on page 5-30 "To delete a database" on page 5-35
Exit	Terminates the execution of Table Designer. Prompts the user to save any changed information that has not been saved.	
Edit menu	Provides the following commands: ◆ Insert Column ◆ Delete Column ◆ Move Up ◆ Move Down ◆ Insert Index ◆ Delete Index ◆ Insert Segment ◆ Delete Segment ◆ Reset ◆ Copy ◆ Paste	

GUI Object	Description	Related Information
Insert Column command or icon	Inserts a column above the cursor location. This command and icon appear only on the Columns tab and Btrieve tab. This command and icon are disabled on the Columns tab when Table Designer is in unlinked mode.	“Modes” on page 5-5 “To change Table Designer mode” on page 5-32 “To insert a column at end of column definitions” on page 5-39 “To insert a column between columns” on page 5-39
Delete Column command or icon	Deletes the column currently selected. This command and icon appear only on the Columns tab and Btrieve tab. This command and icon are disabled on the Columns tab when Table Designer is in unlinked mode.	“Modes” on page 5-5 “To change Table Designer mode” on page 5-32 “To delete a column” on page 5-41
Move Up command or icon	Moves the selected column up. Note that changing column position on the Columns tab affects the physical database. This command and icon are enabled only on the Columns tab and Btrieve tab. This command and icon are disabled on the Columns tab when Table Designer is in unlinked mode.	“Modes” on page 5-5 “To change Table Designer mode” on page 5-32 “To change column order” on page 5-40

GUI Object	Description	Related Information
Move Down command or icon	Moves the selected column down. Note that changing column position on the Columns tab affects the physical database. This command and icon enabled only on the Columns tab and Btrieve tab. This command and icon are disabled on the Columns tab when Table Designer is in unlinked mode.	"Modes" on page 5-5 "To change Table Designer mode" on page 5-32 "To change column order" on page 5-40
Insert Index command or icon	Inserts an index at the cursor location. This command and icon appear only on the Indexes tab.	"To insert an index" on page 5-48
Delete Index command or icon	Deletes the index currently selected. This command and icon appear only on the Indexes tab.	"To delete an index" on page 5-49
Insert Segment command or icon	Inserts a segment for the currently selected index. This command and icon appear only on the Indexes tab.	"To insert an index segment" on page 5-49
Delete Segment command or icon	Deletes the index segment currently selected. This command and icon appear only on the Indexes tab.	"To delete an index segment" on page 5-50
Reset	Restores the information from the last-saved state. All changes that have not been saved are discarded.	"To discard all changes (Reset)" on page 5-37
Copy command or icon	Copies the information in a grid cell to the clipboard.	"To copy data from a grid cell" on page 5-38

GUI Object	Description	Related Information
Paste command or icon	Pastes the clipboard data into the grid cell where the cursor is located.	“To paste data into a grid cell” on page 5-38
View menu	Provides the following commands: ◆ Columns ◆ Indexes ◆ Table Statistics ◆ Btrieve Data ◆ Reset	
Columns	Displays the Columns tab.	“Columns Tab” on page 5-18 “To work with columns” on page 5-32
Indexes	Displays the Indexes tab.	“Indexes Tab” on page 5-20 “To work with indexes” on page 5-32
Table Statistics	Displays the Statistics tab.	“Statistics Tab” on page 5-21 “To display table statistics” on page 5-32

GUI Object	Description	Related Information
Btrieve Data	Displays the Btrieve tab. This command is enabled only if Table Designer is in unlinked mode.	"Btrieve Tab" on page 5-25 "To change Table Designer mode" on page 5-32 "To work with Btrieve data" on page 5-32
Reset command or icon	Restores the information from the last-saved state. All changes that have not been saved are discarded.	"To discard all changes (Reset)" on page 5-37
Tools menu	Provides the following commands: ◆ View Data ◆ Options	
View Data command or icon	Views the data in the table. This icon starts the SQL Data Manager utility.	"To view table data" on page 5-37
Options	Changes Table Designer mode.	"Modes" on page 5-5 "To change Table Designer mode" on page 5-32

GUI Object	Description	Related Information
Help menu	Provides the following commands: ◆ How Do I? ◆ Topics ◆ Search Documentation ◆ What's This? ◆ Glossary ◆ Online Manuals (Web) ◆ Knowledge Base (Web) ◆ Developer Center (Web) ◆ About	
How Do I?	Lists the tasks that you can perform with Table Designer.	"To display help" on page 5-33
Topics	Lists the documentation topics for Table Designer.	"Table Designer Tasks" on page 5-28 "To display help" on page 5-33
Search Documentation	Searches for a string throughout the installed set of Pervasive.SQL documentation.	"To display help" on page 5-33
What's This?	Identifies objects in the Table Designer GUI.	"Table Designer GUI Visual Reference" on page 5-8 "To display help" on page 5-33
Glossary	Defines terms used in the Pervasive.SQL documentation and products.	"To display help" on page 5-33

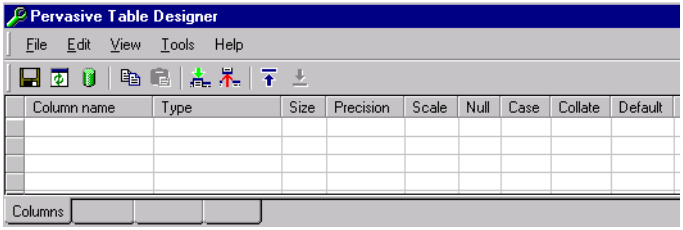
GUI Object	Description	Related Information
Online Manuals (Web)	Directs your browser to www.pervasive.com where you can reference the online manuals.	“To display help” on page 5-33
Knowledge Base (Web)	Directs your browser to www.pervasive.com where you can query the Pervasive.SQL Knowledge Base for additional information about Pervasive.SQL.	“To display help” on page 5-33
Developer Center (Web)	Directs your browser to www.pervasive.com where you can research information specifically for application developers using Pervasive.SQL.	“To display help” on page 5-33
About	Displays the name and version of Table Designer and the Pervasive.SQL copyright information.	“To display help” on page 5-33
Columns tab	Displays the Columns tab. This tab is read-only if Table Designer is in unlinked mode.	“Modes” on page 5-5 “Columns Tab” on page 5-18 “To work with columns” on page 5-32
Indexes tab	Displays the Indexes tab.	“Modes” on page 5-5 “Indexes Tab” on page 5-20 “To work with indexes” on page 5-32

GUI Object	Description	Related Information
Statistics tab	Displays the Statistics tab.	"Modes" on page 5-5 "Statistics Tab" on page 5-21 "To display table statistics" on page 5-32
Btrieve tab	Displays the Btrieve tab. This tab appears only if Table Designer is in unlinked mode.	"Modes" on page 5-5 "Btrieve Tab" on page 5-25 "To change Table Designer mode" on page 5-32 "To work with Btrieve data" on page 5-32
Record length indicator	Shows the number of byes in the current record.	"To determine the length of a record" on page 5-38
Mode indicator	Shows the current mode of Table Designer (linked or unlinked). The mode determines which of the four tabbed dialogs appears.	"Modes" on page 5-5 "To change Table Designer mode" on page 5-32

Columns Tab

The following image shows the Columns tab of the GUI. The table below the image describes the GUI objects. (The menu items and icons are explained in "Main Window" on page 5-8.)

Figure 5-4 Columns Tab



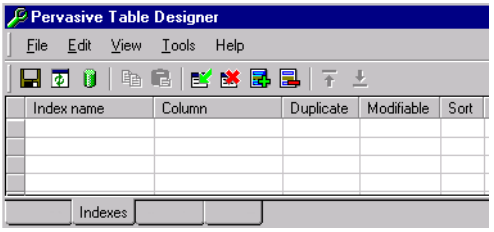
GUI Object	Description	Related Information
Column Name	Specifies the alpha-numeric name for the column. The limit is 20 characters.	"To set a column name" on page 5-41
Type	Specifies the data type of the column.	"Pervasive.SQL Supported Data Types" on page A-2 in *SQL Engine Reference*. "To set a column data type" on page 5-42
Size	Specifies how many bytes are permitted for the data type.	"To set a column size" on page 5-42
Precision	Specifies the number of significant digits for floating point values.	"To set column precision" on page 5-43
Scale	Specifies the number of significant digits that are to the right of the decimal point for floating point values.	"To set a column scale" on page 5-44
Null	Specifies whether NULL values are allowed for the data type. A shaded square (☐) indicates that nulls do not apply to the data type. A checked square (☑) indicates that NULL values are allowed for the data type.	"To set a column to allow or disallow nulls" on page 5-44

GUI Object	Description	Related Information
Case	Specifies whether the database engine uses case-sensitive or case-insensitive comparisons when searching for character values in the database. A checked square (☑) indicates that **case-insensitive** values are used. A shaded square (□) indicates that case sensitivity does not apply to the data type.	"To set case sensitivity for a column" on page 5-45
Collate	Specifies that an alternating collating sequence (ACS) is used for sorting. Contains the path and ACS file name.	"To set a column collating sequence" on page 5-46
Default	Specifies a default value for the column. The default value is used if you perform an SQL INSERT for a row but do not provide a value for the column.	"To set a column default" on page 5-47

Indexes Tab

The following image shows the Indexes tab of the GUI. The table below the image describes the GUI objects. (The menu items and icons are explained in "Main Window" on page 5-8.)

Figure 5-5 Indexes Tab

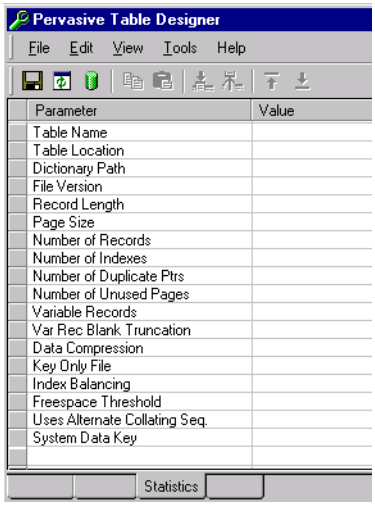


GUI Object	Description	Related Information
Index Name	Specifies the name of the index.	"To insert an index" on page 5-48
Column	Specifies the column used to create the index for sorting.	"To specify index sort column" on page 5-51
Duplicate	Specifies that the index can have duplicate values.	"To allow duplicates in index" on page 5-51
Modifiable	Specifies that the key value can be modified after creation. Allowing modification of key values does not affect performance. Key pages are only updated if the actual key value changes, not if non-key fields in a particular record are changed.	"To specify index as modifiable" on page 5-52
Sort	Specifies the type of sorting to use for the key (ascending or descending).	"To specify index sort sequence" on page 5-53

Statistics Tab

The following image shows the Statistics tab of the GUI. The table below the image describes the GUI objects. (The menu items and icons are explained in "Main Window" on page 5-8.)

Figure 5-6 Statistics Tab



GUI Object	Description	Related Information
Parameter	Labels the statistical information shown on the tabbed area. See the next 18 rows in this table for the labels.	"To display table information" on page 5-54
Table Name	Shows the name of the table as it appears in the database definition.	Naming Conventions in *Pervasive.SQL Programmer's Guide*
Table Location	Shows the physical location of the data file associated with the table. You may edit this parameter to change the location of a table's data file.	"To change the location of a table's data file" on page 5-53
Dictionary Path	Shows where the database's DDF files are located.	
File Version	Shows the earliest Btrieve engine version that can read the file format.	

GUI Object	Description	Related Information
Record Length	Shows the length of the data file's records.	Record Length in *Pervasive.SQL Programmer's Guide*
Page Size	Shows the page size (in bytes) of the data file. The page size determines the maximum number of index segments that can be defined in a table.	Choosing a Page Size in *Pervasive.SQL Programmer's Guide*
Number or Records	Shows the number of records currently contained in the data file.	
Number of Indexes	Shows the number of indexes defined for the table.	Creating Indexes in *Pervasive.SQL Programmer's Guide*
Number of Duplicate Pointers (Ptrs)	Shows the number of linked duplicate indexes that can be added.	File Attributes in *Pervasive.SQL Programmer's Guide*
Number of Unused Pages	Shows the number of pre-allocated pages available. If pre-allocation is enabled, the MicroKernel pre-allocates a specified number of pages when it creates the data file. Pre-allocation guarantees that disk space for the data file is available when the MicroKernel needs it.	
Variable Records	Shows whether the data file contains variable-length records.	Record Length in *Pervasive.SQL Programmer's Guide*

GUI Object	Description	Related Information
Variable Record Blank Truncation	Shows whether blank truncation is enabled. If it is, the MicroKernel truncates the blanks in variable-length records. Blank truncation is applicable only if the Variable Records statistic is Yes and Data Compression is set to No.	Blank Truncation in *Pervasive.SQL Programmer's Guide*
Data Compression	Shows whether data compression is enabled. If it is, the MicroKernel compresses each record it inserts into the data file.	Data Compression in *Pervasive.SQL Programmer's Guide*
Key Only File	Shows the name of the key-only file for the table, if any. A key-only file contains no data records but serves as an index to other Btrieve files.	Key-Only Files in *Pervasive.SQL Programmer's Guide*
Index Balancing	Shows whether balanced indexing is enabled.	Index Balancing in *Pervasive.SQL Programmer's Guide*
FreesSpace Threshold	Shows a percentage (5%, 10%, 20% or 30%) if the data file has a free space threshold. The MicroKernel stores the variable-length portions of records on their own pages (called variable pages), separate from the fixed-length portions (which are stored on data pages). The MicroKernel uses the threshold to determine whether to add data to an existing variable page or to create a new one. A higher free space threshold reduces fragmentation of variable-length records across several pages but uses more disk space.	File Attributes in *Pervasive.SQL Programmer's Guide*

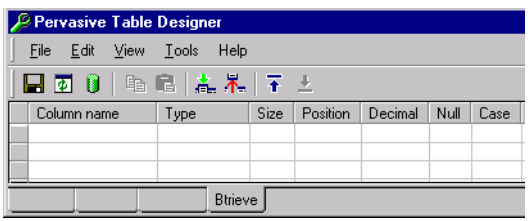
GUI Object	Description	Related Information
Uses Alternate Collating Sequence	Shows whether the table uses an alternate collating sequence for sorting.	Alternate Collating Sequences in *Pervasive.SQL Programmer's Guide*
System Data Key	Shows whether the data file has system data keys enabled.	System Data in *Pervasive.SQL Programmer's Guide*
Value	Shows the value for each parameter.	"To display table information" on page 5-54

Btrieve Tab

The following image shows the Btrieve tab of the GUI. The table below the image describes the GUI objects. (The menu items and icons are explained in "Main Window" on page 5-8.)

Note that this tab appears only when Table Designer is in unlinked mode.

Figure 5-7 Btrieve Tab



GUI Object	Description	Related Information
Column Name	Specifies the alpha-numeric name for the column. The limit is 20 characters.	“To set a Btrieve column name” on page 5-56
Type	Specifies the data type of the column.	“Pervasive.SQL Supported Data Types” on page A-2 in *SQL Engine Reference*. “To set a Btrieve column type”
Size	Specifies how many bytes are permitted for the data type.	“To set a Btrieve column size” on page 5-58
Position	Shows the byte location within the physical file where the column begins.Position is automatically determined by the location of the column and the size of the data type(s) in the preceding column(s).	
Decimal	Specifies the number of significant digits that are to the right of the decimal point for floating point values.	“To set decimal places for a Btrieve column” on page 5-58

GUI Object	Description	Related Information
Null	Specifies whether NULL values are allowed for the data type. A shaded square (☐) indicates that nulls do not apply to the data type. A checked square (☑) indicates that NULL values are allowed for the data type.	"To set a Btrieve column to allow nulls" on page 5-59
Case	Specifies whether the database engine uses case-sensitive or case-insensitive comparisons when searching for character values in the database. A checked square (☑) indicates that **case-insensitive** values are used. A shaded square (☐) indicates that case sensitivity does not apply to the data type.	"To set case sensitivity for a Btrieve column" on page 5-60

Table Designer Tasks

This section explains the tasks that you perform with Table Designer. The tasks are divided into the following categories:

Category	Description
General Tasks	Orient you to the overall use of Table Designer
Columns Tasks	Apply to using the Columns tab
Indexes Tasks	Apply to using the Indexes tab
Statistics Tasks	Apply to using the Statistic tab
Btrieve Tasks	Apply to using the Btrieve tab

General Tasks

General tasks apply to the overall use of the tool. The following table helps you locate a task by category.

Category	Task
Getting Started	“To start Table Designer as a stand-alone application” on page 5-30
	“To start Table Designer from PCC” on page 5-31
	“To identify open database and table” on page 5-31
	“To work with columns” on page 5-32
	“To work with indexes” on page 5-32
	“To display table statistics” on page 5-32
	“To work with Btrieve data” on page 5-32
	“To change Table Designer mode” on page 5-32
	“To display help” on page 5-33
Databases	“To open a database” on page 5-33
	“To create a database” on page 5-34
	“To delete a database” on page 5-35
Tables	“To open a table” on page 5-35
	“To create a table” on page 5-36

Category	Task
	"To delete a table" on page 5-36
Data	"To view table data" on page 5-37
	"To identify insertions or deletions that have not been saved" on page 5-37
	"To save changes" on page 5-37
	"To discard all changes (Reset)" on page 5-37
	"To determine the length of a record" on page 5-38
Copy/Paste	"To copy data from a grid cell" on page 5-38
	"To paste data into a grid cell" on page 5-38

Columns Tasks

Column tasks apply to the Columns tab.

Note Inserting, deleting, or moving a column on the Columns tab changes the database. The database is rebuilt when you save the Table Designer information.

- "To insert a column at end of column definitions" on page 5-39
- "To insert a column between columns" on page 5-39
- "To change column order" on page 5-40
- "To delete a column" on page 5-41
- "To set a column name" on page 5-41
- "To set a column data type" on page 5-42
- "To set a column size" on page 5-42
- "To set column precision" on page 5-43
- "To set a column scale" on page 5-44
- "To set a column to allow or disallow nulls" on page 5-44
- "To set case sensitivity for a column" on page 5-45
- "To set a column collating sequence" on page 5-46
- "To set a column default" on page 5-47
- "To change column order" on page 5-40

Indexes Tasks

Index tasks apply to the Indexes tab.

- "To insert an index" on page 5-48
- "To delete an index" on page 5-49
- "To insert an index segment" on page 5-49
- "To delete an index segment" on page 5-50
- "To specify index sort column" on page 5-51
- "To allow duplicates in index" on page 5-51
- "To specify index as modifiable" on page 5-52
- "To specify index sort sequence" on page 5-53

Statistics Tasks

Statistics tasks apply to the statistics tab.

- "To change the location of a table's data file" on page 5-53
- "To display table information" on page 5-54

Btrieve Tasks

Btrieve tasks apply to the Btrieve tab.

- "To insert a Btrieve column at end of column definitions" on page 5-54
- "To insert a Btrieve column between columns" on page 5-55
- "To delete a Btrieve column" on page 5-56
- "To set a Btrieve column name" on page 5-56
- "To set a Btrieve column type" on page 5-57
- "To set a Btrieve column size" on page 5-58
- "To set decimal places for a Btrieve column" on page 5-58
- "To set a Btrieve column to allow nulls" on page 5-59
- "To set case sensitivity for a Btrieve column" on page 5-60
- "To change Btrieve column position on the GUI" on page 5-60

General Tasks

➢ To start Table Designer as a stand-alone application

1 From the desktop, click **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Table Designer**.

This step assumes that Table Designer was installed as part of a Pervasive.SQL default installation. A vendor application that uses an embedded database engine may require that you start Table Designer with different menu commands. (The GUI executable is named tbldsgn.exe.)

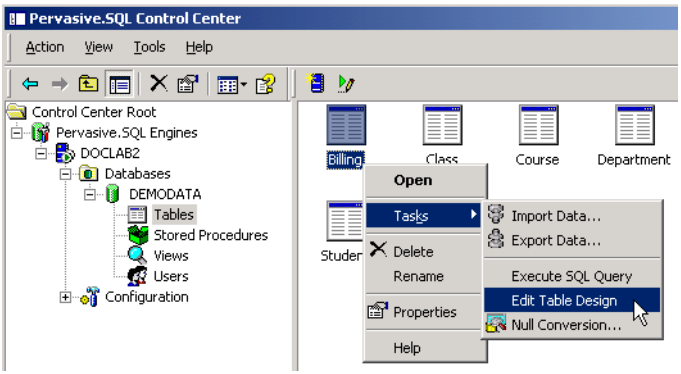
Table Designer starts by providing a dialog in which you choose a database and table to open. By default, the local database engine (on the same computer as Table Designer) is used.

➤ To start Table Designer from PCC

1 Start PCC if it is not already running. (Click **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center.**)

2 In the PCC data pane, right-click on the icon for the table you want to modify.

3 Click **Tasks ▸ Edit Table Design**.



Note When started this way (**Tasks ▸ Edit Table Design**), Table Designer opens a separate instance for each table. This means that you can open multiple instances of Table Designer for the *same* table. Changes saved in one instance are reflected in another instance only if you perform a reset. For example, suppose you open Table Designer twice (called instance A and B) for table MyTable. If you save changes in instance A, you must perform a reset in instance B to see those changes. Reset causes all of the displayed properties to revert to their last-saved values.

➤ To identify open database and table

1 Observe the Table Designer title bar.

The title bar shows the name of the open database and the name of the open table. If Table Designer was started from PCC, the title bar also shows the machine on which the database resides.

For example, the following image shows that Billing is the open table, DEMODATA is the open database, and DOCLAB2 is the machine on which DEMODATA resides.

➤ To work with columns

Ensure that Table Designer is in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click **View** ▸ **Columns** or click the Columns tab.

➤ To work with indexes

When Table Designer is in unlinked mode, index changes affect only the data definition files. See "To change Table Designer mode" on page 5-32.

1 Click **View** ▸ **Indexes** or click the Indexes tab.

➤ To display table statistics

1 Click **View** ▸ **Table Statistics** or click the Statistics tab.

➤ To work with Btrieve data

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32.

1 Click **View** ▸ **Btrieve Data** or click the Btrieve tab.

➤ To change Table Designer mode

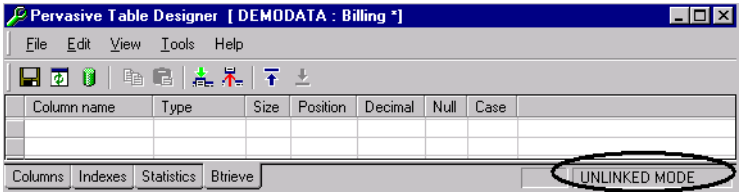
1 Click **Tools** ▸ **Options**.

The Options dialog appears.

2 Deselect the current option. For example, if Table Designer is in linked mode, deselect (uncheck) the option **Table linked mode**.

3 Click **OK** (or click **Apply** then **OK**).

The lower right corner of the Table Designer window indicates the current mode: UNLINKED MODE or LINKED MODE. Note that only in unlinked mode do all four tabbed dialogs appear.



➤ To display help

1 Click **Help**, then the type of help you want:

- **How Do I?** – Lists the tasks that you can perform with Table Designer.
- **Topics** – Lists the documentation topics for Table Designer.
- **Search Documentation** – Searches for a string throughout the installed set of Pervasive.SQL documentation.
- **What's This?** – Identifies objects in the Table Designer GUI.
- **Glossary** – Defines terms used in the Pervasive.SQL documentation and products.
- **Online Manuals (Web)** – Directs your browser to www.pervasive.com where you can reference the online manuals.
- **Knowledge Base (Web)** – Directs your browser to www.pervasive.com where you can query the Pervasive.SQL Knowledge Base for additional information about Pervasive.SQL.
- **Developer Center (Web)** – Directs your browser to www.pervasive.com where you can research information specifically for application developers using Pervasive.SQL.
- **About** – Displays the name and version of Table Designer and the Pervasive.SQL copyright information.

➤ To open a database

These steps apply only if Table Designer is running as a stand-alone application.

1 Click **File ▸ Open Database**.

The Open Database dialog appears that shows a list of databases. (You may click **Create** to create a new database.)

2 Click the database you want in the list.

3 Click **OK**.

If the database has security enabled, a dialog appears on which you must supply a user name and password. Key in the user name (default is Master) and password and click **OK**.

The Open Table dialog appears that shows a list of tables in the database. (You may click **Create** to create a new table.)

4 Click the table you want to open, if applicable. (The list may be empty.)

5 Click **OK**.

➢ To create a database

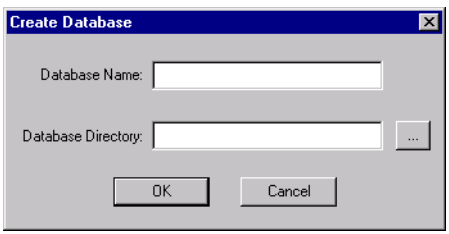
These steps apply only if Table Designer is running as a stand-alone application. To create a database within PCC, see "To add or create a database" on page 3-10.

See also Naming Conventions in *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

You may also create a database from the Open Database dialog. See "To open a database" on page 5-33.

1 Click **File** ▸ **Create Database** (or click **Create** on the Open Database dialog).

The Create Database dialog appears.



2 Type a name for the database in the **Database Name** field.

The name must be unique. An error occurs if the database name already exists.

3 Type a directory location for the database in the **Database Directory** field.

Alternately, you may click the browse button ..., navigate to a directory location, click the location, then click **OK**.

4 Click **OK**.

The Open Database dialog appears that shows a list of databases.

5 Click the desired database to open if it is not already selected.

6 Click **OK**.

The Open Table dialog appears that shows a list of database tables. The list is empty unless you specified a directory location that already contains Pervasive.SQL database tables. If the directory already contains tables, those table names appear in the list.

7 If the list of tables contains names, click the desired table. (You may click **Create** to create a new table.)

8 Click **OK**.

Table Designer creates the database. If a table was selected in step 7, the table is also opened.

Note When you create a database with Table Designer, a .pvs file is created in the same directory location as you specified in step 3. See "PVS File" on page 5-7 for more information.

➤ To delete a database

You cannot delete the current database that you have open in Table Designer.

These steps apply only if Table Designer is running as a stand-alone application. To delete a database within PCC, see "To remove an existing database from an engine" on page 3-13.

1 Click **File ▸ Delete Database**.

The Delete Database dialog appears that shows a list of databases.

2 Click the database you want to delete.

3 Click **OK**.

➤ To open a table

These steps apply only if Table Designer is running as a stand-alone application.

1 Click **File ▸ Open Table**.

The Open Table dialog appears that shows a list of tables for the current database. The list is empty if the current database has no tables. (You may click **Create** to create a new table.)

2 Click the table you want to open.

3 Click **OK**.

➤ To create a table

These steps apply only if Table Designer is running as a stand-alone application. To create a table within PCC, see "To create a new table within a database" on page 3-15.

You may also create a table from the Open Table dialog. See "To open a table" on page 5-35.

1 Click **File ▸ Create Table** (or click **Create** on the Open Table dialog).

The Create Table dialog appears.

2 For **New Table Name**, type the name you want for your table.

An error occurs if you specify a table name that already exists in the database.

3 Click **OK**.

➤ To delete a table

You cannot delete the current table that you have open in Table Designer. That means that you can delete all but one table from your database with Table Designer. (One table must be open to enable the Delete Table command, and you cannot delete the table that is open.) You may also delete tables with the Drop Table wizard. See "Dropping a Table" on page 3-20.

These steps apply only if Table Designer is running as a stand-alone application. To delete a table within PCC, see "To delete an existing table from a database" on page 3-20.

1 Click **File ▸ Delete Table**.

The Delete Table dialog appears that shows a list of tables.

2 Click the table you want to delete.

3 Click **OK**.

➢ To view table data

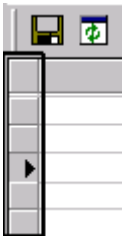
1 Click **Tools** ▸ **View Data** or click .

The command (or the icon) starts Pervasive.SQL SQL Data Manager. For more about SQL Data Manager, see "SQL Data Manager" on page 6-1.

➢ To identify insertions or deletions that have not been saved

1 Observe the cells on the grid indicator column.

The indicator column is the gray, leftmost grid column:



A blue asterisk (*) in a cell shows information that is marked for addition. A red "x" in a cell shows information that is marked for deletion.

➢ To save changes

1 Click **File** ▸ **Save** or click .

2 If Table Designer is in unlinked mode, click **OK** on the warning message to save changes.

➢ To discard all changes (Reset)

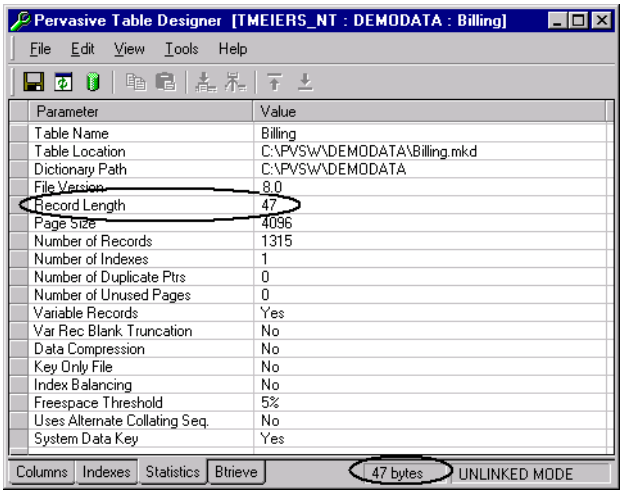
1 Click **View** ▸ **Reset** or click .

Reset restores the table attributes and table data to the last-saved state. All changes that have not been saved are discarded. (Table Designer has three Reset commands. They all perform the same function.)

➤ To determine the length of a record

1 View the record length indicator field near the lower right corner of the Table Designer window.

The value in the field shows the record length in bytes. You may also click the Statistics tab and view the value for the **Record Length** parameter.



➤ To copy data from a grid cell

1 Click in the grid cell that contains the data you want to copy.

2 Click **Edit** ▸ **Copy** or click . (You may also press Control+c.)

➤ To paste data into a grid cell

1 Click in the grid cell where you want to paste the data.

2 Click **Edit** ▸ **Paste** or click . (You may also press Control+v).

Note You cannot paste data that is not applicable to a particular data type. For example, you cannot paste a value into the Size cell for an INTEGER.

You may paste data that you cut or copied from another application, such as a text editor or a spreadsheet program.

Columns Tasks

➤ To insert a column at end of column definitions

Inserting, deleting, or moving a column on the Columns tab changes the database. The database is rebuilt when you save the Table Designer information.

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click **View** ▸ **Columns** or click the Columns tab.

2 Click on the empty grid row below the last named column. You may click on any empty cell in the row except for the gray, leftmost cell. (The gray, leftmost cell is used for positioning on existing columns.)

A new column is inserted. A blue asterisk (*) appears in the leftmost cell. The default name of the column is "col_*n*," where "n" is a number that automatically increments by one. (The first column you insert is col_1, the second col_2 and so forth.) The column is not added to the table until you save the database.

The following image shows a column inserted after the column named Head_of_Dept:

	Column name	Type	Size	Precision	Scale	Null	Case	Collate	Default
	Name	char	20			☐	☑		
	Phone_Number	decimal		11	0	☑	☐		
	Building_Name	char	25			☐	☑		
	Room_Number	uinteger				☐	☐		
	Head_Of_Dept	ubigint				☐	☐		
*	col_1	integer				☑	☐		

3 Click **File** ▸ **Save** or click .

The save action adds the column.

➤ To insert a column between columns

Inserting, deleting, or moving a column on the Columns tab changes the database. The database is rebuilt when you save the Table Designer information.

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click **View** ▸ **Columns** or click the Columns tab.

2 Click on the leftmost cell of the column *above* which you want to insert a column. (A right-pointing triangle appears in the cell, indicating the selected column)

3 Click **Edit** ▸ **Insert Columns** or click

A new column is inserted. A blue asterisk (*) appears in the leftmost cell. The default name of the column is "col_*n*," where "n" is a number that automatically increments by one. (The first column you insert is col_1, the second col_2 and so forth.) The column is not added to the table until you save the database.

The following image shows a column inserted before the column named Head_of_Dept:

	Column name	Type	Size	Precision	Scale	Null	Case	Collate	Default
	Name	char	20			☐	☑		
	Phone_Number	decimal		11	0	☑	☐		
	Building_Name	char	25			☐	☑		
	Room_Number	uinteger				☐	☐		
*	col_1	integer				☑	☐		
	Head_Of_Dept	ubigint				☐	☐		

4 Click **File** ▸ **Save** or click .

The save action adds the column.

➤ To change column order

Inserting, deleting, or moving a column on the Columns tab changes the database. The database is rebuilt when you save the Table Designer information.

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Specify where to move the column.

a. Click **Edit** ▸ **Move Up** (or click)to move the column up one grid row.

b. Click **Edit** ▸ **Move Down** (or click) to move the column down one grid row.

➤ To delete a column

Data is removed from your database if you delete a column. The database is rebuilt when you save the Table Designer information.

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click **Edit** ▸ **Delete Column** or click .

A red "x" appears in the gray, leftmost grid cell . The "x" indicates that the column is marked for deletion. The column is not deleted until you save the table information.

3 Click **File** ▸ **Save** or click .

4 Click **Yes** to save the Table Designer information to catalog files.

The save action deletes the column.

➤ To set a column name

See also Naming Conventions in *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click in the **Column Name** grid cell.

3 Delete the existing column name (use the Backspace key or select the name and press Delete).

4 Type the name you want.

The name can be alpha-numeric, 20 characters or less. The name can have spaces, but if spaces are included you must always enclose the name in quotes whenever using the name in SQL statements. For this reason, Pervasive recommends avoiding spaces. Also avoid using reserved words for column names. See "Reserved Words" in *SQL Engine Reference*.

➤ To set a column data type

The data in your database is converted if you change a column data type. For a listing of data types, see "Pervasive.SQL Supported Data Types" on page A-2 in *SQL Engine Reference*.

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click in the **Type** grid cell.

3 Open the list for data types (click ▾).

4 Scroll to the data type you want.

5 Click the data type you want. (You may also type the first letter of the desired data type to select a data type.)

➤ To set a column size

Data in your database is truncated if you change the column to a smaller size for the following data types:

- CHAR
- NUMERIC

- VARCHAR

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1. Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

   A solid, right-point triangle indicates that the column is selected.

2. Click in the **Size** grid cell.

   You can set a size only for applicable data types, such as CHAR. If size is not applicable, you will be unable to position the cursor in the **Size** grid cell.

3. Delete the existing size (use the Backspace key or select the name and press Delete).

4. Type the size you want.

➤ To set column precision

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1. Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

   A solid, right-point triangle indicates that the column is selected.

2. Click in the **Precision** grid cell.

   You can set precision only for applicable data types, such as DECIMAL. If precision is not applicable, you will be unable to position the cursor in the **Precision** grid cell.

3. Delete the existing value (use the Backspace key or select the name and press Delete).

4. Type the precision value you want.

Precision specifies the number of significant digits for floating point values.

➤ To set a column scale

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click in the **Scale** grid cell.

You can set a scale value only for applicable data types, such as NUMERIC. If collating sequence is not applicable, you will be unable to position the cursor in the **Scale** grid cell.

3 Delete the existing value (use the Backspace key or select the name and press Delete).

4 Type the scale value you want.

Scale specifies the number of significant digits that are to the right of the decimal point for floating point values.

➤ To set a column to allow or disallow nulls

For additional information about nulls, see "Null Conversion" on page 15-2 in *Advanced Operations Guide*, "INSERT" on page 3-85 in *SQL Engine Reference*, and Null Value in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click the option box in the **Null** grid cell.

You can allow nulls only for applicable data types. A shaded square indicates that null values do not apply to the data type.

A checkmark in the box indicates that nulls **are** allowed for the column. Lack of a checkmark indicates that the column cannot be blank (that is, for every record, the column must contain a value other than null).

Option Box State	Meaning
☑	Nulls allowed
☐	Nulls disallowed
▪	Nulls do not apply

Also see "Null Values" on page 5-6.

➤ To set case sensitivity for a column

Case sensitivity does not apply if the key uses an alternate collating sequence (ACS). You cannot specify case sensitivity and use an ACS.

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click the option box in the **Case** grid cell.

You can set a collating sequence only for applicable data types. A shaded square indicates that case sensitivity does not apply to the data type.

A checkmark in the box indicates that column values are case **insensitive**. Lack of a checkmark indicates that column values are case sensitive.

Option Box State	Meaning
☑	Case insensitive
☐	Case sensitive
▪	Case does not apply

By default, Pervasive.SQL is case sensitive when sorting string keys. Uppercase letters are sorted before lowercase letters. If you specify case insensitive, values are sorted without distinguishing case.

➤ To set a column collating sequence

For additional information about collating sequences, see "Manipulating Btrieve Data Files with Maintenance" on page 14-1 in *Advanced Operations Guide* and Alternate Collating Sequences in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

If you use an alternate collating sequence (ACS), you cannot specify case sensitivity. Case sensitivity does not apply if the key uses an ACS.

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click in the **Collate** grid cell.

You can set an alternating collating sequence (ACS) only for applicable data types. If collating sequence is not applicable, you will be unable to position the cursor in the **Collate** grid cell.

3 Delete the existing value, if present (use the Backspace key or select the name and press Delete).

4 Type the path and ACS file name you want.

Pervasive.SQL supplies an ACS file, upper.alt, in the Samples folder. To use this file, you would type `c:\pvsw\samples\upper.alt`.

Upper.alt treats upper and lower case letters the same for sorting. For example, if a database has values abc, ABC, DEF, and Def, inserted in that order, the sorting with upper.alt returns as abc, ABC, DEF, and Def. (The values abc and ABC, and the values DEF and Def are considered duplicates and are returned in the order in which they were inserted.) Normal ASCII sorting sequences upper case letters before lower case, such that the sorting would return as ABC, DEF, Def, abc.

➤ To set a column default

This task requires that Table Designer be in linked mode. See "To change Table Designer mode" on page 5-32.

1 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

2 Click in the **Default** grid cell.

You can set a default only for applicable data types. If a default is not applicable, you will be unable to position the cursor in the the **Default** grid cell.

3 Delete the existing value, if present (use the Backspace key or select the name and press Delete).

4 Type the default value you want.

The default value is used if you perform an SQL INSERT for a row but do not provide a value for the column.

Index Tasks

➤ To insert an index

When Table Designer is in unlinked mode, index changes affect only the data definition files.

Tip You may create "dummy" indexes in your data definition files to match indexes in your Btrieve file. An example helps clarify.

Suppose your Btrieve file has indexes 0, 1, 2, and 5 and in your Index.ddf file you have indexes 0, 1, and 2. You could create "dummy" indexes in the definition file for 3 and 4, and an actual index for 5. Thus, your indexes would match as follows:

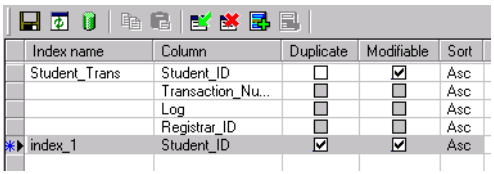
Btrieve File Indexes	Data Definition File Indexes
0	0
1	1
2	2
	3 "dummy"
	4 "dummy"
5	5

The "dummy" indexes can be on any column that can be indexed because they are not used by your Btrieve file.

1. Click **Edit ▸ Indexes** or click the Indexes tab.
2. Click on the empty grid row below the last named index. (You may also click on the gray, leftmost cell of an existing index if you want to insert a new index above an existing one.)

   A new index is inserted. A blue asterisk (*) appears in the leftmost gray cell. The default name of the column is "index_*n*," where "n" is a number that automatically increments by one. (The first index you insert is index_1, the second index_2 and so forth.)

The following image shows an index inserted:



3 Click **File** ▸ **Save** or click .

The save action adds the index. See also Creating Indexes in *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

➢ To delete an index

When Table Designer is in unlinked mode, index changes affect only the data definition files.

1 Click on the grid indicator cell for the index you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the index is selected.

2 Click **Edit** ▸ **Delete Index** or click .

A red "x" appears in the gray, leftmost grid cell . The "x" indicates that the index is marked for deletion. The index is not deleted until you save the table information.

3 Click **File** ▸ **Save** or click .

The save action deletes the index.

➢ To insert an index segment

For detailed information about segments, see Segmentation in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

When Table Designer is in unlinked mode, index changes affect only the data definition files.

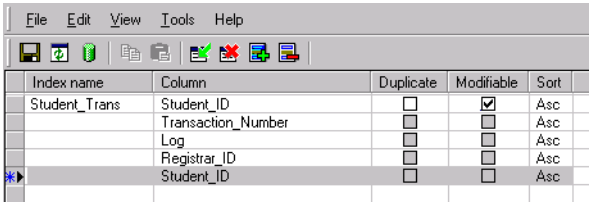
1 Click **View** ▸ **Indexes** or click the Indexes tab.

2 Click on the leftmost cell of an existing index or an existing index segment.

3 Click **Edit** ▸ **Insert Segment** or click .

A new index segment is inserted below the existing index or segment. A blue asterisk (*) appears in the leftmost cell. The new segment is not added until you save the database.

The following image shows an index segment inserted:



4 Click **File** ▸ **Save** or click .

The save action adds the index segment.

➢ To delete an index segment

When Table Designer is in unlinked mode, index changes affect only the data definition files.

1 Click **Edit** ▸ **Indexes** or click the Indexes tab.

2 Click on the grid indicator cell for the index segment you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the index segment is selected.

3 Click **Edit** ▸ **Delete Segment** or click .

A red "x" appears in the gray, leftmost grid cell . The "x" indicates that the index segment is marked for deletion. The segment is not deleted until you save the table information.

4 Click **File** ▸ **Save** or click .

The save action deletes the index segment

➤ To specify index sort column

For detailed information about sort order, see Sort Order in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

When Table Designer is in unlinked mode, index changes affect only the data definition files.

1. Click **Edit** ▸ **Indexes** or click the Indexes tab.
2. Click in the **Column** grid cell.
3. Open the list for columns (click).
4. Scroll to the column name you want and click the name.

   Some data types, such as BIT, cannot be used for an index. Columns with such data types are not valid choices.

➤ To allow duplicates in index

For detailed information about duplicates, see Duplicatability in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

When Table Designer is in unlinked mode, index changes affect only the data definition files.

1. Click **Edit** ▸ **Indexes** or click the Indexes tab.
2. Click on the grid indicator cell for the index you want. The indicator cell is the gray, leftmost grid cell:

   A solid, right-point triangle indicates that the index is selected.
3. Click the option box in the **Duplicate** grid cell.

   A checkmark in the box indicates that the index allows duplicates (the values are not unique).

Lack of a checkmark indicates that the index values are unique (duplicate values are not allowed).

Option Box State	Meaning
☑	Duplicates allowed
☐	Duplicates not allowed
▪	Duplicates do not apply

The default for all SQL data types is that the index column allows duplicates.

➤ To specify index as modifiable

For detailed information about modifiability, see Modifiability in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

When Table Designer is in unlinked mode, index changes affect only the data definition files.

1. Click **Edit ▸ Indexes** or click the Indexes tab.
2. Click on the grid indicator cell for the index you want. The indicator cell is the gray, leftmost grid cell:

   A solid, right-point triangle indicates that the index is selected.
3. Click the option box in the **Modifiable** grid cell.

   A checkmark in the box indicates that the index value can be modified. Lack of a checkmark indicates that the index value cannot be modified.

Option Box State	Meaning
☑	Values can be modified
☐	Values cannot be modified
▪	Modifiable does not apply

The default for all SQL data types is that the index column is modifiable.

➤ To specify index sort sequence

For detailed information about sort order, see Sort Order in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

When Table Designer is in unlinked mode, index changes affect only the data definition files.

1. Click **Edit ▸ Indexes** or click the Indexes tab.
2. Click in the **Sort** grid cell.

   If a sort sequence is not applicable, you will be unable to position the cursor in the **Sort** grid cell.
3. Open the list for sort (click).
4. Click the type of sort you want: ascending (asc) or descending (des). The default value is ascending.

Statistics Tasks

➤ To change the location of a table's data file

Changing the Table Location parameter does **not** move the table's data file. The change only instructs the database engine where to find the data file. If you wish to have your existing data still available, you must move the data file to the newly specified location.

A database also must be unbound for you to change the location of a data file or move the data file. When you create a database in PCC, you are offered the option of creating a bound database.

Pervasive.SQL automatically binds a data file to a named database if the file meets any of the following criteria:

- Is part of a bound database
- Has a trigger

- Has a foreign key
- Has a primary key that is referenced by a foreign key.

See also the tool "Maintain Named Databases" on page 5-12 in *Advanced Operations Guide*. This tool lets you bind and unbind a database.

1 Click **View ▸ Table Statistics** or click the Statistics tab.

2 Click in the **Table Location** grid cell and delete the existing value, if present. (Use the Backspace key or select the name and press Delete.)

3 Type the location you want.

If you are specifying a location on the same server where the database engine is running, specify the location in this manner:

- For NetWare, enter a path in the form *vol*:*path*.
- For Windows NT, enter a path in the form *drive*:*path*, where *drive* is the local drive letter.
- For Linux, enter the standard Linux path format from root.

If you are specifying a location to data files on another server, specify the full UNC path:

- For NetWare, enter the path in the form *server**vol*:*path*
- For Windows, enter the path in the form *server**sharename**path*
- For Linux, enter the path in one of two ways:

  *server**share**path* or

  server\$PVSW$*path*

➢ To display table information

1 Click **View ▸ Table Statistics** or click the Statistics tab.

Btrieve Tasks

➢ To insert a Btrieve column at end of column definitions

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1 Click **View** ▸ **Btrieve Data** or click the Btrieve tab.

2 Click on the empty grid row below the last named column. You may click on any empty cell in the row except for the leftmost cell. (The leftmost cell is used for positioning on existing columns.)

A new column is inserted. A blue asterisk (*) appears in the leftmost cell. The default name of the column is "col_*n*," where "n" is a number that automatically increments by one. (The first column you insert is col_1, the second col_2 and so forth.) The column is not added to the definition files until you save the database.

The following image shows a column inserted after the column named Head_of_Dept:

	Column name	Type	Size	Precision	Scale	Null	Case	Collate	Default
	Name	char	20			☐	☑		
	Phone_Number	decimal		11	0	☑	☐		
	Building_Name	char	25			☐	☑		
	Room_Number	uinteger				☐	☐		
	Head_Of_Dept	ubigint				☐	☐		
* ▸	col_1	integer				☑	☐		

3 Click **File** ▸ **Save** or click .

The save action adds the column to the definition files.

➢ To insert a Btrieve column between columns

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1 Click **View** ▸ **Btrieve Data** or click the Btrieve tab.

2 Click on the leftmost cell of the column *above* which you want to insert a column. (A right-pointing triangle appears in the cell, indicating the selected column)

3 Click **Edit** ▸ **Insert Columns** or click

A new column is inserted. A blue asterisk (*) appears in the leftmost cell. The default name of the column is "col_*n*," where "n" is a number that automatically increments by one. (The first column you insert is col_1, the second col_2 and so forth.) The column is not added to the definition files until you save the database.

The following image shows a column inserted before the column named Head_of_Dept:

	Column name	Type	Size	Precision	Scale	Null	Case	Collate	Default
	Name	char	20			☐	☑		
	Phone_Number	decimal		11	0	☑	☐		
	Building_Name	char	25			☐	☑		
	Room_Number	uinteger				☐	☐		
* ▸	col_1	integer				☑	☐		
▸	Head_Of_Dept	ubigint				☐	☐		

4 Click **File** ▸ **Save** or click .

The save action adds the column to the definition files.

➢ To delete a Btrieve column

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1 Click **View** ▸ **Btrieve Data** or click the Btrieve tab.

2 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

3 Click **Edit** ▸ **Delete Column** or click .

A red "x" appears in the gray, leftmost grid cell . The "x" indicates that the column is marked for deletion. The column is not deleted until you save the table information.

4 Click **File** ▸ **Save** or click .

The save action deletes the column from the definition files.

➢ To set a Btrieve column name

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1 Click **View** ▸ **Btrieve Data** or click the Btrieve tab.

2 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

3 Click in the **Column Name** grid cell.

4 Delete the existing column name (use the Backspace key or select the name and press Delete).

5 Type the name you want.

The name can be alpha-numeric, 20 characters or less.

➤ To set a Btrieve column type

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1 Click **View** ▸ **Btrieve Data** or click the Btrieve tab.

2 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

3 Click in the **Type** grid cell.

4 Open the list for data types (click).

5 Scroll to the data type you want.

6 Click the data type you want.

For a listing of data types, see "Pervasive.SQL Supported Data Types" on page A-2 in *SQL Engine Reference*.

➤ To set a Btrieve column size

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1. Click **View** ▸ **Btrieve Data** or click the Btrieve tab.
2. Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

   A solid, right-point triangle indicates that the column is selected.
3. Click in the **Size** grid cell.

   You can set a size only for applicable data types, such as CHAR. If size is not applicable, you will be unable to position the cursor in the **Size** grid cell.
4. Delete the existing size (use the Backspace key or select the name and press Delete).
5. Type the size you want.

➤ To set decimal places for a Btrieve column

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1. Click **View** ▸ **Btrieve Data** or click the Btrieve tab.
2. Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

   A solid, right-point triangle indicates that the column is selected.
3. Click in the **Decimal** grid cell.

You can set a decimal value only for applicable data types.

4 Delete the existing value (use the Backspace key or select the name and press Delete).

5 Type the decimal value you want.

Decimal specifies the number of significant digits that are to the right of the decimal point for floating point values.

➤ To set a Btrieve column to allow nulls

Ensure that Table Designer is in unlinked mode. See "To change Table Designer mode" on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

For detailed information about null values, see Null Value in the *Pervasive.SQL Programmer's Guide*, which is part of the Pervasive.SQL Software Developer's Kit (SDK).

1 Click **View** ▸ **Btrieve Data** or click the Btrieve tab.

2 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

3 Click the option box in the **Null** grid cell.

You can allow nulls only for applicable data types. A shaded square indicates that null values do not apply to the data type.

A checkmark in the box indicates that nulls **are** allowed for the column. Lack of a checkmark indicates that the column cannot be blank (that is, for every record, the column must contain a value other than null).

Option Box State	Meaning
☑	Nulls allowed
☐	Nulls disallowed
▪	Nulls do not apply

Also see “Null Values” on page 5-6.

➤ To set case sensitivity for a Btrieve column

Ensure that Table Designer is in unlinked mode. See “To change Table Designer mode” on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1 Click **View ▸ Btrieve Data** or click the Btrieve tab.

2 Click on the grid indicator cell for the column you want. The indicator cell is the gray, leftmost grid cell:

A solid, right-point triangle indicates that the column is selected.

3 Click the option box in the **Case** grid cell.

You can set a collating sequence only for applicable data types. A shaded square indicates that case sensitivity does not apply to the data type.

A checkmark in the box indicates that column values are case **insensitive**. Lack of a checkmark indicates that column values are case sensitive.

Option Box State	Meaning
☑	Case insensitive
☐	Case sensitive
▪	Case does not apply

➤ To change Btrieve column position on the GUI

Ensure that Table Designer is in unlinked mode. See “To change Table Designer mode” on page 5-32. Tasks performed on the Btrieve tab affect only the table definition files. The data files are not modified.

1 Click **View ▸ Btrieve Data** or click the Btrieve tab.

2 Specify where to position the column.

a. Click **Edit** ▸ **Move Up** (or click)to move the column up one grid row.

b. Click **Edit** ▸ **Move Down** (or click) to move the column down one grid row.

Note that moving a column up or down changes the position only on the Table Designer GUI and in the data definition files. The data remains unaffected.

chapter

6

SQL Data Manager

A Tour of Pervasive SQL Data Manager

The topics in this chapter include:

- "SQL Data Manager Concepts" on page 6-2
- "SQL Data Manager GUI Visual Reference" on page 6-16
- "SQL Data Manager Tasks" on page 6-40

SQL Data Manager Concepts

This section contains the following topics:

- "Overview" on page 6-2
- "Restrictions" on page 6-3
- "Results Tabs" on page 6-4
- "Panes" on page 6-8
- "SQL Reference" on page 6-15

Overview

SQL Data Manager (SQL DM) allows you to run Structured Query Language (SQL) statements against a Pervasive.SQL database. With SQL statements, you may retrieve, create, change, or delete data in a database provided you have the proper database permissions to perform these actions. The SQL statements that you may use with SQL DM are documented in *SQL Engine Reference*.

SQL DM includes a graphical user interface (GUI) that runs only on Windows 32-bit platforms. You may start the GUI as a stand-alone application or from within Pervasive Control Center (PCC).

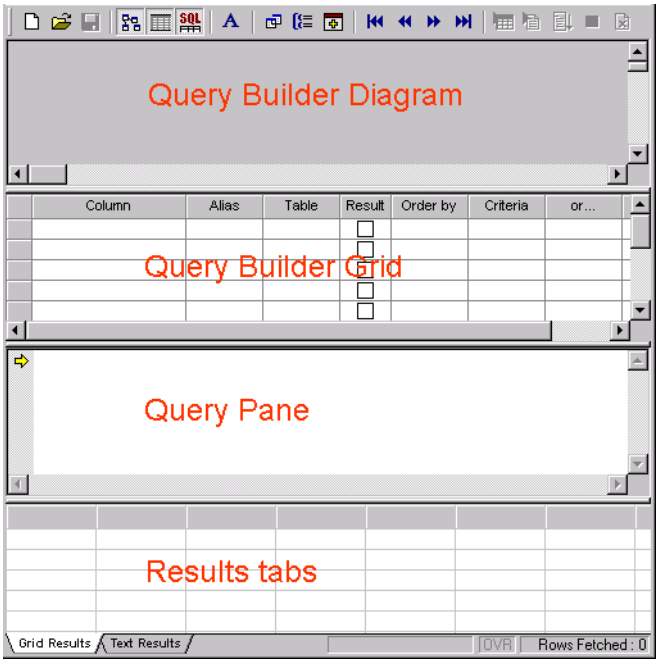
Caution Backup all your data definition files (DDFs) and data files before you perform functions through SQL DM. This tool gives you the ability to modify your database table definitions and data. You could inadvertently change your files in an irreversible manner. Full recovery is possible if you have performed a backup.

SQL DM contains four primary work areas (see Figure 6-1):

- Query Builder Diagram
- Query Builder Grid
- Query Pane
- Results tabs

This section discusses the primary areas in detail. See "SQL Data Manager GUI Visual Reference" on page 6-16 to identify other objects in the GUI, such as menu commands, toolbar icons, and so forth.

Figure 6-1 Primary Areas of SQL DM Graphical User Interface



Restrictions

The following actions and SQL statements are not supported in SQL DM:

- Creating a database
- Use of dynamic parameters (indicated by a question mark)
- COMMIT and START TRANSACTION
- SET SECURITY.

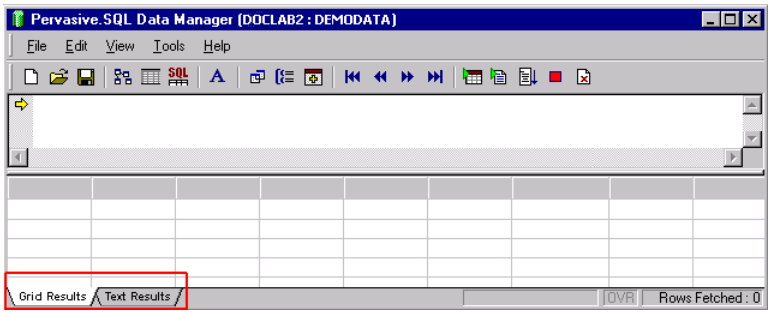
Results Tabs

SQL DM contains two tabbed views that are used to display the results of running SQL statements:

- Grid Results
- Text Results

The tabbed views always appear; you may select one or the other but cannot hide them.

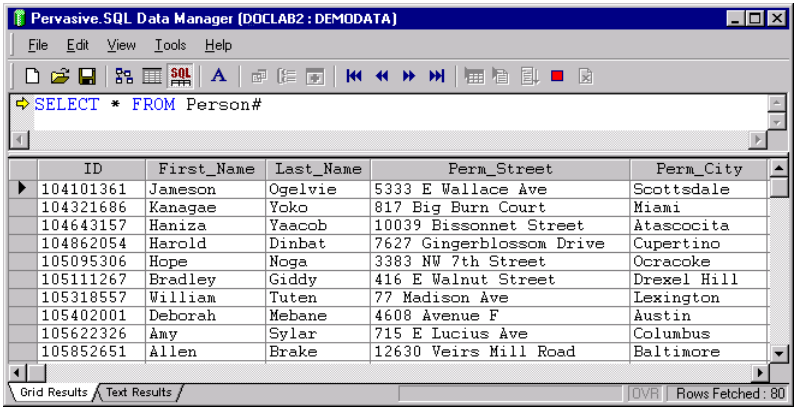
Figure 6-2 SQL DM Results Tabs



Grid Results Tab

The Grid Results tab shows in a matrix format, like a spreadsheet, the result of running SQL statements. Each field is represented as a column and the data appears in cells within the columns. The grid is used only for the results of SELECT statements (statements that return data).

Figure 6-3 Grid Results Tabbed Dialog

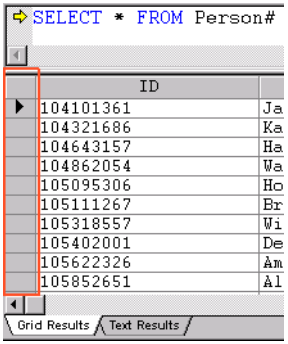
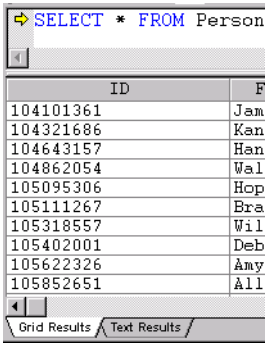

Pervasive.SQL Data Manager (DOCLAB2 : DEMODATA)
File Edit View Tools Help
SELECT * FROM Person#
ID First_Name Last_Name Perm_Street Perm_City
104101361 Jameson Ogelvie 5333 E Wallace Ave Scottsdale
104321686 Kanagae Yoko 817 Big Burn Court Miami
104643157 Haniza Yaacob 10039 Bissonnet Street Atascocita
104862054 Harold Dinbat 7627 Gingerblossom Drive Cupertino
105095306 Hope Noga 3383 NW 7th Street Ocracoke
105111267 Bradley Giddy 416 E Walnut Street Drexel Hill
105318557 William Tuten 77 Madison Ave Lexington
105402001 Deborah Mebane 4608 Avenue F Austin
105622326 Amy Sylar 715 E Lucius Ave Columbus
105852651 Allen Brake 12630 Veirs Mill Road Baltimore
Grid Results Text Results
OVR Rows Fetched : 80

Grid States

The results grid has two states: active and display-only. The active state allows you to directly change database data by changing the values in the grid cells. You must have the proper table permissions to affect the database data. See "To add a new user to the database" on page 3-25, which explains how to set permissions for a user.

The active state is indicated by a gray indicator column on the very left of the grid.

Table 6-1 Comparison of Active and Display-only Grid

Active Grid (indicator column on left)	Display-only Grid
	

The indicator column uses icons to inform you which row is active (contains the cursor) or whether data has changed in that row.

Table 6-2 Icons in Results Grid Indicator Column

Active Row	Cell Value(s) Has Changed In Row

When the grid is in the display-only state, you cannot change the values in grid cells. The display-only state also lacks the indicator column. By default, SQL DM starts with the Results Grid as the active tab and the grid in the active state.

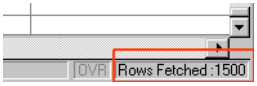
For tasks that allow you to change the grid state, see "To set Grid Results to allow changes to database data" on page 6-58 and "To set default behavior for Grid Results tab" on page 6-59.

Note If you have a table without primary keys that contains records with duplicate data for a column, you will be unable to update the duplicate data on the Grid Results tab. SQL DM cannot tell one record from another because of the duplicate data. This problem does not occur if your table contains primary keys because they prevent duplicate data.

Records Affected and Scrolling

Regardless of state, the grid caches results locally and initially displays as many records as the vertical size of the grid permits. The larger you size the main window and the larger the vertical size of the grid within it, the more records the grid can initially display.

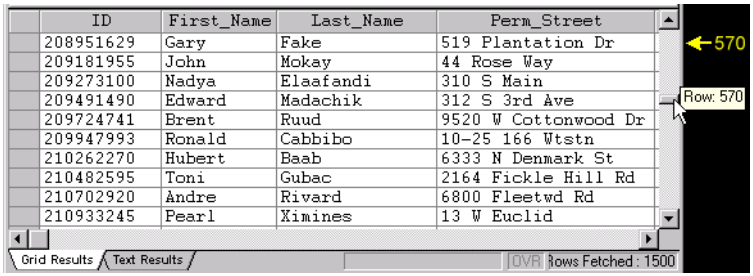
As you scroll the vertical scroll bar, more records are made available to the grid. The number of records retrieved appears in the lower right corner of the main window (in the "rows fetched" indicator).



Once you scroll to the bottom, the rows fetched indicator reports the total number of records returned by the SQL statement.

Once scrolled to the bottom and released, the scroll box then displays a tool tip for all subsequent scrolling. The tool tip tells you which record is positioned at the top of the visible grid. For example, the following figure shows that record 570 in the current table is the top line of the visible grid.

Figure 6-4 Scroll Box Tool Tip Indicating Row at Top of Grid


ID
First_Name
Last_Name
Perm_Street
208951629 Gary Fake 519 Plantation Dr
209181955 John Mokay 44 Rose Way
209273100 Nadya Elaafandi 310 S Main
209491490 Edward Madachik 312 S 3rd Ave
209724741 Brent Ruud 9520 W Cottonwood Dr
209947993 Ronald Cabbibo 10-25 166 Wtstn
210262270 Hubert Baab 6333 N Denmark St
210482595 Toni Gubac 2164 Fickle Hill Rd
210702920 Andre Rivard 6800 Fleetwd Rd
210933245 Pearl Ximines 13 W Euclid
570
Row: 570
Grid Results
Text Results
OVR
Rows Fetched : 1500

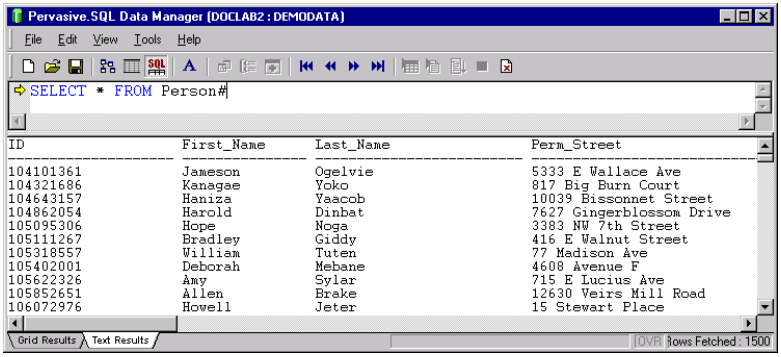
Text Results Tab

The Text Results tab shows in a text format the result of running SQL statements. You cannot change the data values in the database by changing the text.

The Text Results tab is automatically used for the results of any SQL statement that is not a SELECT statement. For example, suppose you want to delete some records and have typed a DELETE statement into Query Pane. If you select the menu command **Run Current SQL Statement in Grid**, SQL DM returns the results to the Text Results tab, *not* to the Grid Results tab.

You may also use the Text Results tab for SELECT statements to show data returned. The data returned appears in a columnar format with each field represented as an underlined heading. The length of the underline (number of characters) reflects the field width. The data appears as rows below the headings, left aligned with each heading.

Figure 6-5 Text Result Tab View


Pervasive.SQL Data Manager (DOCLAB2 : DEMODATA)
File Edit View Tools Help
SELECT * FROM Person#
ID First_Name Last_Name Perm_Street
104101361 Jameson Ogelvie 5333 E Wallace Ave
104321686 Kanagae Yoko 817 Big Burn Court
104643157 Haniza Yaacob 10039 Bissonnet Street
104862054 Harold Dinbat 7627 Gingerblossom Drive
105095306 Hope Noga 3383 NW 7th Street
105111267 Bradley Giddy 416 E Walnut Street
105318557 William Tuten 77 Madison Ave
105402001 Deborah Mebane 4608 Avenue F
105622326 Amy Sylar 715 E Lucius Ave
105852651 Allen Brake 12630 Veirs Mill Road
106072976 Howell Jeter 15 Stewart Place
Grid Results Text Results
Rows Fetched : 1500

If you stop the execution of SQL statements, or if execution stops because of an error, the Text Results tab lists the statement that was last run. Knowing the last statement run can help you troubleshoot problems. See "To stop running SQL statements" on page 6-57 and "To continue running SQL statements that were stopped" on page 6-57.

Panes

SQL DM contains three panes that you use as the primary work areas (see Figure 6-1):

- Query Pane
- Query Builder Diagram
- Query Builder Grid

Unlike the tabbed dialogs, each pane may be individually shown or hidden. Each pane may be sized vertically within the SQL DM main window.

Together, Query Builder Diagram and Query Builder Grid compose what is called *Query Builder.*

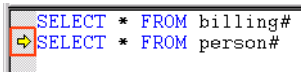
Query Pane

Query Pane is a text entry area for SQL statements. You may type directly in the pane, add SQL statements to the pane from Query Builder, or combine both methods. By default, text does not wrap within Query Pane. You may turn on word wrap if you choose. See "To wrap text within Query Pane" on page 6-52.

If you add more than one SQL statement in Query Pane, you must separate each statement with a delimiter. By default, SQL DM uses

the pound sign (#). You may change this delimiter character to the semicolon (;). See "To set SQL statement separator" on page 6-51. Note that you must use the pound sign as the statement delimiter in Query Pane if you create stored procedures. Stored procedures use the semicolon as a statement delimiter within the procedure itself.

You may run the current statement or all statements in Query Pane. See "To run a single SQL statement" on page 6-56 and "To run all SQL statements" on page 6-57. The statement in which the cursor is located is the current statement, and is indicated by a yellow arrow in the left margin.



Comments within Query Pane are indicated by double dashes (--). See "To type a comment into Query Pane" on page 6-52.

By default, SQL DM starts with Query Pane shown.

Cursor Movement Within Query Pane

You have two options when you move the cursor from statement to statement within Query Pane:

- Move the cursor only (with the mouse or keyboard arrow keys). See "To move the cursor within Query Pane without running any SQL statements" on page 6-49).
- Move the cursor (with the toolbar icons) **and** run the SQL statement to which it moves. See "To move the cursor within Query Pane and run the SQL statement to which it moves" on page 6-50.

The second option is a convenience that lets you perform two actions with one mouse click. (This option is particularly handy if you want to "step" through SQL statements.)

Query Builder

Query Builder allows you to create SQL statements (database queries) without directly writing SQL. Query Builder comprises Query Builder Diagram and Query Builder Grid.

You will find Query Builder most useful if you are new to writing SQL or if you want to check a syntactically correct form of a statement. If you are experienced with writing SQL, you will

probably prefer to type your statements directly into Query Pane. Query Builder Diagram works best for creating relatively simple queries.

You may create the following types of SQL statements with Query Builder:

- SELECT (see "To build a SELECT statement with Query Builder" on page 6-63)
- UPDATE (see "To build an UPDATE statement with Query Builder" on page 6-76)
- INSERT (see "To build an INSERT statement with Query Builder" on page 6-79)
- DELETE (see "To build a DELETE statement with Query Builder" on page 6-82).

A SELECT statement retrieves data from one or more tables. An UPDATE statement changes the values of columns in one or more existing rows in a table. An INSERT statement creates a new row in a table and inserts values into specified columns. A DELETE statement removes one or more rows from a table.

Mixing Use of Typed Statements and Built Statements

You may mix the use of statements that you type into Query Pane and statements added by Query Builder. For example, you could begin a statement with Query Builder, then manually type additional syntax. Note, however, that Query Builder is not aware of what you add manually to Query Pane. This means that the next time Query Builder adds to the statement being build, your manually added portion will be removed.

An example helps clarify this. Suppose you use Query Builder to build a simple SELECT statement. You then type a WHERE clause into Query Pane. The statement is valid and may be run (provided the syntax you typed is correct). If you then use Query Builder to continue building your statement, your manually typed WHERE clause will be removed.

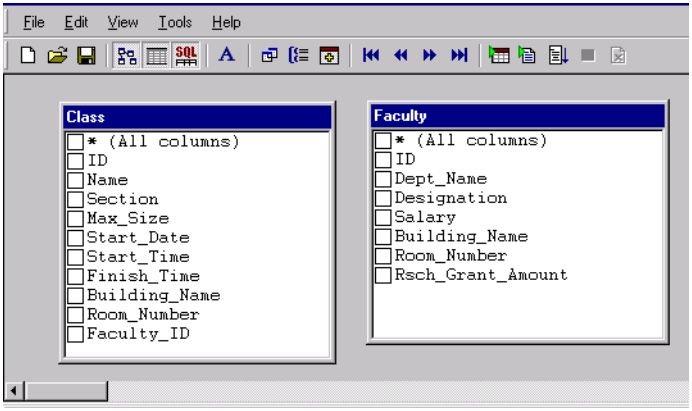
If you wish to keep what is in Query Pane, you may save the statements to an ASCII file (called a query script). See "To save an SQL query as a script" on page 6-47 and "To open an SQL script" on page 6-48.

Query Builder Diagram

Query Builder Diagram allows you to open tables and select columns to which the SQL statements apply. If shown, Query Builder Diagram always appears as the first pane below the main window toolbar. (See Figure 6-1 on page 6-3.) By default, when you start SQL DM, only Query Pane is shown (and the Grid Results tab is the active tab).

Query Builder Diagram represents a table as a window within the Diagram pane. The title bar of the window contains the name of the table. For example, Figure 6-6 shows that tables "Class" and "Faculty," from the DEMODATA sample database, have been added to Query Builder Diagram.

Figure 6-6 Query Builder Diagram with Tables Class and Faculty



Each table window shows the columns within that table. For SELECT, UPDATE, and INSERT statements, the window contains check boxes that allow you select columns. The following table explains the symbols used within a table window.

Table 6-3 Symbols Used Within a Table Window

Symbol	Explanation
✔	Marks a check box for a SELECT statement. Indicates that a table column is to appear in Query Builder Grid.
	Marks a check box for an UPDATE statement. Indicates that a table column is to appear in Query Builder Grid.

Table 6-3 Symbols Used Within a Table Window

Symbol	Explanation
	Marks a check box for an INSERT statement. Indicates that a table column is to appear in Query Builder Grid.
	Indicates that all columns are unselectable for a DELETE statement.You cannot delete individual columns so check boxes do not appear in the table window for a DELETE statement. Instead of selecting columns, you specify the criteria in Query Builder Grid for the rows you want to delete.
	Indicates that the column is being ordered by ascending values. This symbol appears if you select ascending for a column on Query Builder Grid.
	Indicates that the column is being ordered by descending values. This symbol appears if you select descending for a column on Query Builder Grid.
	Indicates that the column is used in a GROUP BY clause.

Behavior When Query Builder Diagram Contains No Tables

If Query Builder Diagram contains no tables (such as the first time you show Query Builder Diagram for a particular session of SQL DM), the following occurs:

1. Query Builder Diagram appears
2. Query Builder Grid appears
3. The Add Table dialog appears.

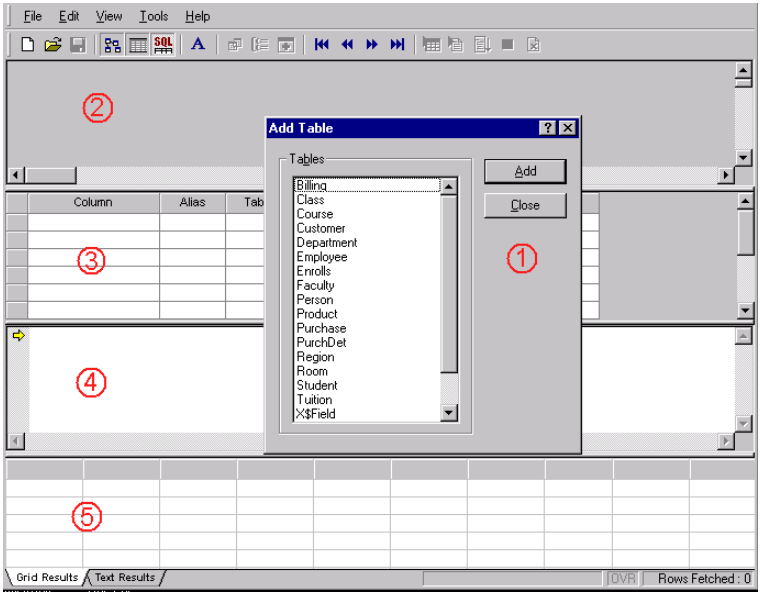
Actions 2 and 3 occur as a convenience. Query Builder Diagram works in tandem with Query Builder Grid. The Grid lets you further specify the conditions that create the SQL statement. Query Builder Diagram also requires tables, so it makes sense to let you select the initial tables with which you want to work.

If Query Builder Diagram contains one or more tables, Query Builder Diagram and Query Builder Grid must be shown (or hidden) independently. See “To show or hide Query Builder Diagram” on page 6-62 and “To show or hide Query Builder Grid” on page 6-62.

The following image shows what occurs if you show Query Builder Diagram for the sample database DEMODATA and Query Builder Diagram contains no tables:

1. The Add Table dialog appears.
2. Query Builder Diagram appears.
3. Query Builder Grid appears, showing the grid for a SELECT statement.
4. Query Pane appears.
5. Grid Results tab is active.

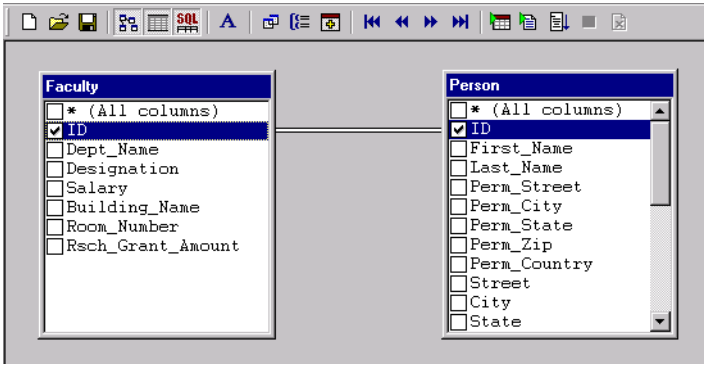
Figure 6-7 Results of Showing Query Builder Diagram Containing No Tables



Query Builder Diagram allows you to create an inner join between tables by clicking on a column in one table and dragging the column to a corresponding column in another table. A join line then appears between the two table windows. Each pair of columns connected in this manner contains its own join line, so two tables may show one or more join lines. Note that the SQL statement created uses a WHERE clause and not the JOIN syntax.

The following image shows a join line between columns “ID” in tables “Faculty” and “Person.”

Figure 6-8 Example of Join Line Between Columns



Query Builder Grid

Query Builder Grid lets you refine an SQL statement based on the tables and selected columns in Query Builder Diagram. The Grid allows you to specify query options for the SQL statements, such as which columns to include in the results, how to order the results, the criteria for the statement, and so forth.

Query Builder Grid always appears as the first pane below Query Builder Diagram (if shown). See Figure 6-1 on page 6-3. If Query Builder Diagram is hidden, Query Builder Grid appears as the first pane below the main window toolbar.

Each of the four types of SQL statement that you can build with Query Builder uses a Query Builder Grid specific for that type. Each grid contains input columns appropriate for the type of statement being built. For an explanation of the input columns for each grid, see the following:

- "Query Builder Grid for SELECT Statements" on page 6-32
- "Query Builder Grid for UPDATE Statements" on page 6-35
- "Query Builder Grid for INSERT Statements" on page 6-36
- "Query Builder Grid for DELETE Statements" on page 6-37.

For an explanation of using each grid to create an SQL statement, see the following:

- "To build a SELECT statement with Query Builder" on page 6-63
- "To build an UPDATE statement with Query Builder" on page 6-76

- "To build an INSERT statement with Query Builder" on page 6-79
- "To build a DELETE statement with Query Builder" on page 6-82.

Behavior of Query Builder Grid When Query Builder Diagram Contains No Tables

The same behavior occurs as explained in "Behavior When Query Builder Diagram Contains No Tables" on page 6-12.

SQL Reference

The SQL statements and key words that you may use with SQL DM are documented in *SQL Engine Reference*.

SQL Data Manager GUI Visual Reference

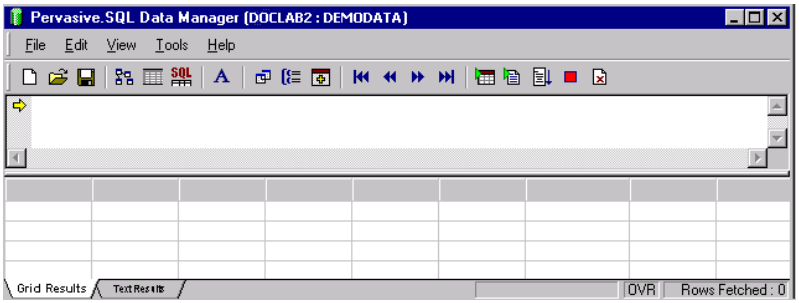
This section describes the objects on the SQL DM graphical user interface (GUI).

- Main Window on page 6-16
- Query Builder Diagram on page 6-31
- Query Builder Grid for SELECT Statements on page 6-32
- Query Builder Grid for UPDATE Statements on page 6-35
- Query Builder Grid for INSERT Statements on page 6-36
- Query Builder Grid for DELETE Statements on page 6-37

Main Window

The following image shows the main window of the GUI. The table below the image describes the GUI objects. By default, Query Builder Diagram and Query Builder Grid are hidden when you start SQL DM.

Figure 6-9 SQL Data Manager Main Window



GUI Object	Description	Related Information
Title Bar	Shows the following information: ◆ name of the utility ◆ name of the computer on which the database resides (if a database is open) ◆ name of the open database (if one is open) (For example, in Figure 6-9 the open database is DEMODATA on machine DOCLAB2.)	
File menu	Provides the following commands: ◆ New SQL Query ◆ Open SQL Query ◆ Save SQL Query ◆ Save SQL Query As ◆ Open Database ◆ Close Database ◆ Print ◆ Page Setup ◆ Exit	

GUI Object	Description	Related Information
New SQL Query command or icon	Creates a new SQL script in Query Pane. Clears Query Pane and positions the statement indicator at the top of the pane.	"Query Pane" on page 6-8 "To create an SQL query or script" on page 6-44
Open SQL Query command or icon	Opens an SQL script and places its contents into Query Pane.	"Query Pane" on page 6-8 "To open an SQL script" on page 6-48
Save SQL Query command or icon	Saves the SQL statement(s) in Query Pane as an ASCII file called an SQL script.	"Query Pane" on page 6-8 "To save an SQL query as a script" on page 6-47
Save SQL Query As	Saves the SQL statement(s) in Query Pane as an ASCII file called an SQL script. If you have an SQL script already open, this command lets you save the statements with a different file name.	"Query Pane" on page 6-8 "To save an SQL query as a script" on page 6-47
Open Database	Opens a database for use within SQL DM. If a database is already open, SQL DM closes it and opens the one you specify.	"To open a database" on page 6-44
Close Database	Closes the database currently open in SQL DM.	"To close a database" on page 6-45
Print	Allows you to specify options for printing the contents of Query Pane.	"Query Pane" on page 6-8 "To print the contents of Query Pane" on page 6-55

GUI Object	Description	Related Information
Page Setup	Allows you to specify options for the layout of the printed page.	"To layout the printed page" on page 6-55
Exit	Terminates the execution of SQL DM. Prompts you to save the contents of Query Pane if the contents have changed.	"To exit SQL DM" on page 6-49
Edit menu	Provides the following commands: ◆ Undo (Control+Z) ◆ Cut (Control+X) ◆ Copy (Control+C) ◆ Paste (Control+V) ◆ Select All (Control+A) ◆ Find (Control+F) ◆ Find Next (F3) ◆ Replace (Control+H)	
Undo (Control+Z)	Cancels typing actions performed in Query Pane. May be repeatedly performed for multiple undo actions.	"Query Pane" on page 6-8 "To cancel (undo) typing actions in Query Pane" on page 6-53 "To cancel an Undo action (redo) in Query Pane" on page 6-53
Cut (Control+X)	Removes selected text from Query Pane and places it in the clipboard.	"Query Pane" on page 6-8 "To cut or delete text from Query Pane" on page 6-55

GUI Object	Description	Related Information
Copy (Control+C)	Copies selected text from Query Pane and places it in the clipboard.	"Query Pane" on page 6-8 "To copy text from Query Pane" on page 6-54
Paste (Control+V)	Pastes clipboard text into Query Pane at cursor position.	"Query Pane" on page 6-8 "To paste within Query Pane" on page 6-55
Select All (Control+A)	Selects all of the contents of Query Pane.	"Query Pane" on page 6-8 "To select text in Query Pane" on page 6-54
Find (Control+F)	Finds a string in Query Pane.	"Query Pane" on page 6-8 "To find text in Query Pane" on page 6-53
Find Next (F3)	Searches for the last string specified for the Find actions.	"Query Pane" on page 6-8 "To find text in Query Pane" on page 6-53
Replace (Control+H)	Replaces one string with another in Query Pane.	"Query Pane" on page 6-8 "To replace text in Query Pane" on page 6-54

GUI Object	Description	Related Information
View menu	Provides the following commands: ◆ Query Pane ◆ Query Builder Diagram ◆ Query Builder Grid ◆ Set Font ◆ Word Wrap	
Query Pane command or icon	Shows or hides Query Pane. By default, Query Pane is shown when you start SQL DM. A database must be open to enable this command or icon.	“Query Pane” on page 6-8 “To show or hide Query Pane” on page 6-49
Query Builder Diagram command or icon	Shows or hides Query Builder Diagram. A database must be open to enable this command or icon.	“Query Builder Diagram” on page 6-11 “To show or hide Query Builder Diagram” on page 6-62 “To build a SELECT statement with Query Builder” on page 6-63 “To build an UPDATE statement with Query Builder” on page 6-76 “To build an INSERT statement with Query Builder” on page 6-79 “To build a DELETE statement with Query Builder” on page 6-82

GUI Object	Description	Related Information
Query Builder Grid command or icon	Shows or hides Query Builder Grid. A database must be open to enable this command or icon.	"Query Builder Grid" on page 6-14 "To show or hide Query Builder Grid" on page 6-62 "To build a SELECT statement with Query Builder" on page 6-63 "To build an UPDATE statement with Query Builder" on page 6-76 "To build an INSERT statement with Query Builder" on page 6-79 "To build a DELETE statement with Query Builder" on page 6-82
Set Font command or icon	Set the font family, style, size, and script. The font characteristics apply to all text in Query Pane. By default, SQL DM sets the font characteristics to Courier Regular 10, Western script.	"Query Pane" on page 6-8 "To set the font for Query Pane" on page 6-56
Word Wrap	Wraps text within Query Pane. A checkmark beside the command indicates that word wrap is active.	"To wrap text within Query Pane" on page 6-52
Tools menu	Provides the following commands: ◆ Query ◆ Query Builder ◆ Active Grid ◆ Properties	

GUI Object	Description	Related Information
Query	Provides the following commands: ◆ Run Current SQL Statement in Grid ◆ Run Current SQL Statement in Text ◆ Run All SQL Statements ◆ Stop ◆ Continue ◆ Clear	
Run Current SQL Statement in Grid command or icon	Runs the current statement in Query Pane and places the results on the Grid Results tab if the statement is a SELECT statement. If the current statement is not a SELECT statement, the results are placed on the Text Results tab. This command is enabled if a database is open and one or more SQL statements are present in Query Pane.	"Results Tabs" on page 6-4 "Query Pane" on page 6-8 "To set Grid Results to allow changes to database data" on page 6-58 "To set SQL statement separator" on page 6-51 "To run a single SQL statement" on page 6-56
Run Current SQL Statement in Text command or icon	Runs the current statement in Query Pane and appends the results on the Text Results tab. This command is enabled if a database is open and one or more SQL statements are present in Query Pane.	"Text Results Tab" on page 6-7 "Query Pane" on page 6-8 "To set SQL statement separator" on page 6-51 "To run a single SQL statement" on page 6-56

GUI Object	Description	Related Information
Run All SQL Statements command or icon	Runs all statements in Query Pane starting with the statement at the top. Output is placed on the Text Results tab, never on the Grid Results tab. The output is appended to any results that already exists on the Text Results tab. This command is enabled if a database is open and one or more SQL statements are present in Query Pane.	“Text Results Tab” on page 6-7 “Query Pane” on page 6-8 “To set SQL statement separator” on page 6-51 “To run all SQL statements” on page 6-57
Stop command or icon	Interrupts the current statement(s) being run. The statement indicator positions to the statement in Query Pane that was being run when the interruption occurred. In addition, the Text Results tab lists the statement that was last run. If you are running a statement for output to the Grid Results tab, any results are cleared on the tab. An SQL statement(s) must be running to enable this command.	“To set a query time-out period” on page 6-51 “To stop running SQL statements” on page 6-57 “To continue running SQL statements that were stopped” on page 6-57
Continue	Continues running a script that was interrupted with the Stop command or Stop icon, or that stopped running because of an error.	“To stop running SQL statements” on page 6-57 “To continue running SQL statements that were stopped” on page 6-57
Clear command or icon	Clears all results on the Text Results tab. This icon is active only if the Text Results tab contains output. A database must be open to enable this command or icon.	“To clear results from Text Results tab” on page 6-61

GUI Object	Description	Related Information
Query Builder	Provides the following commands: ◆ Change Query Type ◆ Add Table ◆ Group By	"To show or hide Query Builder Diagram" on page 6-62
Change Query Type command or icon	Changes the type of statement being built with Query Builder. The choices for type are SELECT, INSERT, UPDATE, and DELETE. This command is enabled if a database is open and Query Builder Diagram or Query Builder Grid is enabled.	"Query Builder Grid" on page 6-14 "To set query type for Query Builder" on page 6-62
Add Table command or icon	Adds a table to Query Builder Diagram (so you may select fields from the table). This command is enabled if a database is open and Query Builder Diagram or Query Builder Grid is enabled.	"Query Builder Diagram" on page 6-11 "Query Builder Grid" on page 6-14 "To add a table to Query Builder Diagram" on page 6-86 "To remove a table from Query Builder Diagram" on page 6-87
Group By command or icon	Specifies a GROUP BY clause in a SELECT statement when you build an SQL statement with Query Builder. This command is enabled if a database is open and Query Builder Diagram or Query Builder Grid is enabled.	"Query Builder Diagram" on page 6-11 "Query Builder Grid" on page 6-14 "To add a Group By column to Query Builder Grid" on page 6-88 "To remove a table from Query Builder Diagram" on page 6-87

GUI Object	Description	Related Information
Active Grid	Sets the Grid Results grid to active or display-only. An active grid allows you to change values in the cells, which changes the data values in the database provided you have the proper database permissions. By default, the grid is active.	"To set default behavior for Grid Results tab" on page 6-59
Properties	Sets the default behavior of the Grid Results grid, the SQL statement delimiter and the query time-out value.	"To set default behavior for Grid Results tab" on page 6-59 "To set SQL statement separator" on page 6-51 "To set a query time-out period" on page 6-51
Help menu	Provides the following commands: ◆ How Do I? ◆ Topics ◆ SQL Reference ◆ Search Documentation ◆ What's This? ◆ Glossary ◆ Online Manuals (Web) ◆ Knowledge Base (Web) ◆ Developer Center (Web) ◆ About Pervasive.SQL Data Manager	
How Do I?	Lists the tasks that you can perform with SQL DM.	"SQL Data Manager Tasks" on page 6-40 "To display help" on page 6-48

GUI Object	Description	Related Information
Topics	Lists the documentation topics pertaining to SQL DM.	“To display help” on page 6-48
SQL Reference	Displays an alphabetical list of SQL statements and keywords that are links to the *SQL Engine Reference*.	
Search Documentation	Searches for a string throughout the installed set of Pervasive.SQL documentation.	“To display help” on page 6-48
What’s This?	Identifies objects in the SQL DM GUI.	“SQL Data Manager GUI Visual Reference” on page 6-16 “To display help” on page 6-48
Glossary	Defines terms used in the Pervasive.SQL documentation.	“To display help” on page 6-48
Online Manuals (Web)	Directs your browser to www.pervasive.com where you can reference the online manuals.	“To display help” on page 6-48
Knowledge Base (Web)	Directs your browser to www.pervasive.com where you can query the Pervasive.SQL Knowledge Base for additional information about Pervasive.SQL.	“To display help” on page 6-48
Developer Center (Web)	Directs your browser to www.pervasive.com where you can research information specifically for application developers using Pervasive.SQL.	“To display help” on page 6-48

GUI Object	Description	Related Information
About Pervasive.SQL Data Manager	Displays the name and version of SQL DM and the Pervasive.SQL copyright information.	"To display help" on page 6-48
	Moves the cursor to the first line in Query Pane and runs that statement. The statement indicator, the yellow arrow in the left margin of Query Pane, indicates the statement in which the cursor is located.	"Query Pane" on page 6-8 "To move the cursor within Query Pane without running any SQL statements" on page 6-49 "To move the cursor within Query Pane and run the SQL statement to which it moves" on page 6-50 "To set SQL statement separator" on page 6-51
	Moves the statement indicator to the previous line in Query Pane and runs that statement. The statement indicator, the yellow arrow in the left margin of Query Pane, indicates the statement in which the cursor is located.	"Query Pane" on page 6-8 "To move the cursor within Query Pane without running any SQL statements" on page 6-49 "To move the cursor within Query Pane and run the SQL statement to which it moves" on page 6-50 "To set SQL statement separator" on page 6-51

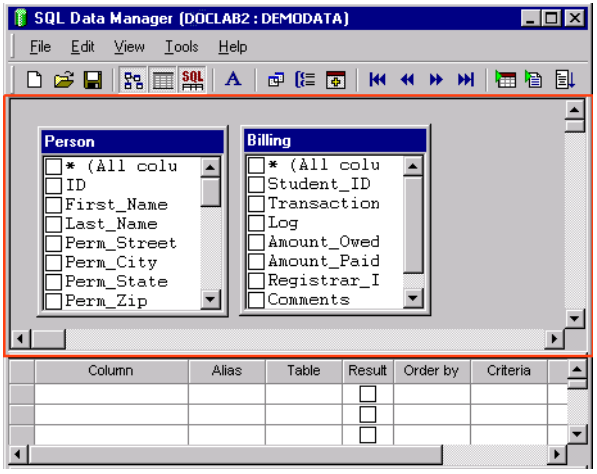
GUI Object	Description	Related Information
	Moves the statement indicator to the next line in Query Pane and runs that statement. The statement indicator, the yellow arrow in the left margin of Query Pane, indicates the statement in which the cursor is located.	"Query Pane" on page 6-8 "To move the cursor within Query Pane without running any SQL statements" on page 6-49 "To move the cursor within Query Pane and run the SQL statement to which it moves" on page 6-50 "To set SQL statement separator" on page 6-51
	Moves the statement indicator to the last line in Query Pane and runs that statement. The statement indicator, the yellow arrow in the left margin of Query Pane, indicates the statement in which the cursor is located.	"Query Pane" on page 6-8 "To move the cursor within Query Pane without running any SQL statements" on page 6-49 "To move the cursor within Query Pane and run the SQL statement to which it moves" on page 6-50 "To set SQL statement separator" on page 6-51
Query Pane	Accepts SQL statements typed into the pane or added with Query Builder. By default, Query Pane is shown (it may be shown or hidden).	"Query Pane" on page 6-8 "To create an SQL query or script" on page 6-44 "To show or hide Query Pane" on page 6-49

GUI Object	Description	Related Information
Results Grid	Shows in a spreadsheet-like format the output of SELECT statements.	“Grid Results Tab” on page 6-4 “To set SQL statement separator” on page 6-51 “To set Grid Results to allow changes to database data” on page 6-58 “To copy data from Grid Results tab to clipboard” on page 6-60
Grid Results tab	Displays the Results Grid tabbed dialog.	“Grid Results Tab” on page 6-4 “To work with grid results” on page 6-58
Text Results tab	Displays the Text Results tabbed dialog.	“Text Results Tab” on page 6-7 “To work with text results” on page 6-61
Progress indicator	Displays a progress bar (a series of moving blue squares) to indicate that SQL DM is currently processing an action. The progress bar stops displaying after an action completes.	

GUI Object	Description	Related Information
Insert/Overtype indicator	Shows whether Query Pane and the grid on the Grid Results tab are in insert or overtype mode. Insert mode adds characters between other characters as you type. Existing characters are pushed to the right. Overtype mode replaces characters as you type. The default mode is insert. When this field shows **OVR** in boldface, SQL DM is in overtype mode. When OVR shows in a dimmed typeface, SQL DM is in insert mode.	"Query Pane" on page 6-8 "To toggle insert/overtype mode" on page 6-56
Rows fetched indicator	Displays the number of rows returned for a SELECT statement.	"Records Affected and Scrolling" on page 6-6

Query Builder Diagram

The following image shows the pane for Query Builder Diagram. Two tables, "Person" and "Billing," are open in the pane. When shown, Query Builder Diagram always appears as the first pane below the main window toolbar.



Query Builder Grid for SELECT Statements

The following image shows Query Builder Grid for SELECT statements. The table below the image describes the GUI objects.

Figure 6-10 Query Builder Grid for SELECT Statements

GUI Object	Description	Related Information
Column	Shows the name of the database column being used in the SQL statement. Column names are fully qualified with the table name (table_name.column_name).	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63
Alias	Shows the alternate name for a column. The alias name appears in an AS clause. The alias also appears as the column title in the results on the Grid Results tab and the Text Results tab.	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63
Table	Shows the name of the table with which the column is associated.	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63
Result	Specifies whether to include the column in Query Pane. An "X" in the option square includes the column in the SQL statement. Lack of an "X" excludes the column.	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63

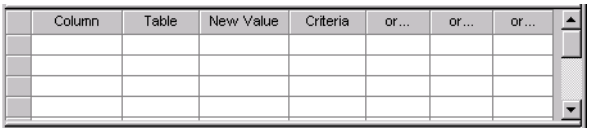
GUI Object	Description	Related Information
Order by	Specifies the priority by which the columns will be sorted in the result. The following choices are available: ◆ Ascending ◆ Descending ◆ Unsorted The default is unsorted.	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63
Group by	Specifies that the column is being used in a GROUP BY clause or as part of an aggregate query. The Group By column appears only if you click **Tools ▸ Query Builder ▸ Group By** or click . Group By provides the following aggregate functions: ◆ Group By (the default) ◆ AVG (average) ◆ COUNT ◆ MAX (maximum) ◆ MIN (minimum) ◆ SUM	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63

GUI Object	Description	Related Information
Criteria	Specifies a search condition for the associated column. The default operator is equals (=). Text literals are automatically single quoted when you move from the Criteria cell. The criteria acts as a filter to restrict the number of rows a SELECT statement returns. The following operators are recognized as search criteria for a value: BETWEEN *expression* AND *expression* NOT BETWEEN expression AND expression IN NOT IN IS NULL IS NOT NULL LIKE NOT LIKE = (equals) < (less than) > (greater than) <= (less than or equal to) >= (greater than or equal to) <> (not equal to)	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63
Or	Specifies an additional search condition linked to the previous criteria by a logical OR. **Note**: The grid does not contain a column for a logical AND. The task "To build a SELECT statement with Query Builder" explains how to construct a logical AND.	"Query Builder Grid" on page 6-14 "To build a SELECT statement with Query Builder" on page 6-63

Query Builder Grid for UPDATE Statements

The following image shows Query Builder Grid for UPDATE statements. The table below the image describes the GUI objects.

Figure 6-11 Query Builder Grid for UPDATE Statements



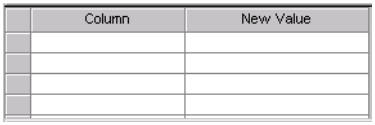
GUI Object	Description	Related Information
Column	Shows the name of the database column being used in the SQL statement. Column names are fully qualified with the table name (table_name.column_name).	"Query Builder Grid" on page 6-14 "To build an UPDATE statement with Query Builder" on page 6-76
Table	Shows the name of the table with which the column is associated.	"Query Builder Grid" on page 6-14 "To build an UPDATE statement with Query Builder" on page 6-76
New Value	Specifies the value used to replace (update) a value in an existing column.	"Query Builder Grid" on page 6-14 "To build an UPDATE statement with Query Builder" on page 6-76

GUI Object	Description	Related Information
Criteria	Specifies a search condition for the associated column. The default operator is equals (=). Text literals are automatically single quoted when you move from the Criteria cell. The criteria specifies the data that you want to change. The following operators are recognized as search criteria for a value: BETWEEN expression AND expression NOT BETWEEN expression AND expression IN NOT IN IS NULL IS NOT NULL LIKE NOT LIKE = (equals) < (less than) > (greater than) <= (less than or equal to) >= (greater than or equal to) <> (not equal to)	"Query Builder Grid" on page 6-14 "To build an UPDATE statement with Query Builder" on page 6-76
Or	Specifies an additional search condition linked to the previous criteria by a logical OR. **Note**: The grid does not contain a column for a logical AND. The task "To build a SELECT statement with Query Builder" explains how to construct a logical AND.	"Query Builder Grid" on page 6-14 "To build an UPDATE statement with Query Builder" on page 6-76 "To build a SELECT statement with Query Builder" on page 6-63

Query Builder Grid for INSERT Statements

The following image shows Query Builder Grid for INSERT statements. The table below the image describes the GUI objects.

Figure 6-12 Query Builder Grid for INSERT Statements

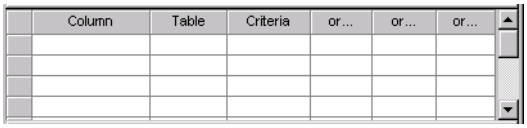


GUI Object	Description	Related Information
Column	Shows the name of the database column being used in the SQL statement. Column names are fully qualified with the table name (table_name.column_name).	"Query Builder Grid" on page 6-14 "To build an INSERT statement with Query Builder" on page 6-79
New Value	Specifies the value to be inserted into an existing column.	"Query Builder Grid" on page 6-14 "To build an INSERT statement with Query Builder" on page 6-79

Query Builder Grid for DELETE Statements

The following image shows Query Builder Grid for DELETE statements. The table below the image describes the GUI objects.

Figure 6-13 Query Builder Grid for DELETE Statement



GUI Object	Description	Related Information
Column	Shows the name of the database column being used in the SQL statement. Column names are fully qualified with the table name (table_name.column_name).	"Query Builder Grid" on page 6-14 "To build a DELETE statement with Query Builder" on page 6-82
Table	Shows the name of the table with which the column is associated.	"Query Builder Grid" on page 6-14 "To build a DELETE statement with Query Builder" on page 6-82

GUI Object	Description	Related Information
Criteria	Specifies a search condition for the associated column. The default operator is equals (=). Text literals are automatically single quoted when you move from the Criteria cell. The criteria specifies the data that is used to identify the rows to delete. Note that a DELETE statement removes the entire row in which the criteria is found. If you want to change data and not remove it, use an UPDATE statement. The following operators are recognized as search criteria for a value: BETWEEN expression AND expression NOT BETWEEN expression AND expression IN NOT IN IS NULL IS NOT NULL LIKE NOT LIKE = (equals) < (less than) > (greater than) <= (less than or equal to) >= (greater than or equal to) <> (not equal to)	"Query Builder Grid" on page 6-14 "To build a DELETE statement with Query Builder" on page 6-82
Or	Specifies an additional search condition linked to the previous criteria by a logical OR. **Note**: The grid does not contain a column for a logical AND. The task "To build a SELECT statement with Query Builder" explains how to construct a logical AND.	"Query Builder Grid" on page 6-14 "To build a DELETE statement with Query Builder" on page 6-82 "To build a SELECT statement with Query Builder" on page 6-63

SQL Data Manager Tasks

This section explains the tasks that you perform with SQL DM. The tasks are divided into the following categories:

Category	Description
General Tasks	Orient you to the overall use of SQL DM
Query Pane Tasks	Help you use Query Pane
Execution and Results Tasks	Apply to the running of SQL statements and working with the results
Query Builder Tasks	Orient you to the use of Query Builder Diagram and Query Builder Grid

What to do first? If you are new to SQL DM, begin with the general tasks. If you are new to SQL programming, begin with the tasks for Query Builder. If you are experienced with SQL programming, begin with the tasks for Query Pane.

General Tasks

General tasks orient you to the overall use of SQL DM.

- "To start SQL DM as a stand-alone application" on page 6-42
- "To start SQL DM from PCC" on page 6-43
- "To identify machine and database currently open" on page 6-43
- "To open a database" on page 6-44
- "To create an SQL query or script" on page 6-44
- "To close a database" on page 6-45
- "To change data with Grid Results grid" on page 6-45
- "To save an SQL query as a script" on page 6-47
- "To open an SQL script" on page 6-48
- "To display help" on page 6-48
- "To exit SQL DM" on page 6-49

Query Pane Tasks

Query Pane tasks help you use Query Pane.

- "To show or hide Query Pane" on page 6-49
- "To move the cursor within Query Pane without running any SQL statements" on page 6-49

- "To move the cursor within Query Pane and run the SQL statement to which it moves" on page 6-50
- "To set SQL statement separator" on page 6-51
- "To wrap text within Query Pane" on page 6-52
- "To cancel (undo) typing actions in Query Pane" on page 6-53
- "To find text in Query Pane" on page 6-53
- "To replace text in Query Pane" on page 6-54
- "To select text in Query Pane" on page 6-54
- "To copy text from Query Pane" on page 6-54
- "To paste within Query Pane" on page 6-55
- "To cut or delete text from Query Pane" on page 6-55
- "To print the contents of Query Pane" on page 6-55
- "To layout the printed page" on page 6-55
- "To toggle insert/overtype mode" on page 6-56
- "To set the font for Query Pane" on page 6-56

Execution and Results Tasks

Execution and results tasks apply to the running of SQL statements and working with results on the Grid Results tab and the Text Results tab.

- "To run a single SQL statement" on page 6-56
- "To run all SQL statements" on page 6-57
- "To stop running SQL statements" on page 6-57
- "To work with grid results" on page 6-58
- "To set Grid Results to allow changes to database data" on page 6-58
- "To set default behavior for Grid Results tab" on page 6-59
- "To delete a row(s) from Grid Results tab" on page 6-59
- "To work with text results" on page 6-61
- "To clear results from Text Results tab" on page 6-61
- "To select text on Text Results tab" on page 6-61
- "To copy text from Text Results tab" on page 6-61
- "To cut or delete text from Text Results tab" on page 6-61

Query Builder Tasks

Query Builder Diagram tasks orient you to the use of Query Builder Diagram and Query Builder Grid.

- "To show or hide Query Builder Diagram" on page 6-62
- "To show or hide Query Builder Grid" on page 6-62
- "To set query type for Query Builder" on page 6-62
- "To build a SELECT statement with Query Builder" on page 6-63
- "To build an UPDATE statement with Query Builder" on page 6-76
- "To build an INSERT statement with Query Builder" on page 6-79
- "To build a DELETE statement with Query Builder" on page 6-82
- "To add a table to Query Builder Diagram" on page 6-86
- "To remove a table from Query Builder Diagram" on page 6-87
- "To insert a row into Query Builder Grid" on page 6-87
- "To delete a row(s) from Query Builder Grid" on page 6-88
- "To delete a row(s) from Query Builder Grid" on page 6-88
- "To add a Group By column to Query Builder Grid" on page 6-88

General Tasks

➢ To start SQL DM as a stand-alone application

1 From the desktop, click **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ SQL Data Manager**.

This step assumes that SQL DM was installed as part of a Pervasive.SQL default installation. A vendor application that uses an embedded database engine may require that you start SQL DM with different menu commands. (The GUI executable is named sqldmgr.exe.)

SQL DM starts by providing a dialog in which you choose a database engine and database name to connect to. By default, the local database engine (on the same computer as SQL DM) is specified.

2 Click the desired database engine and database to open.

3 Click **OK**.

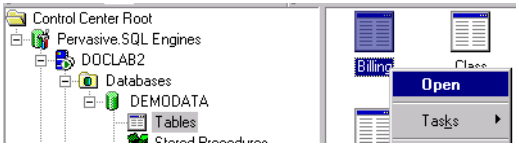
Note SQL DM opens a separate instance for each database. This means that you can open multiple instances of SQL DM for the *same* database. Changes saved in one instance are reflected in another instance only if you run a query (or queries) in the other instance. For example, suppose you start SQL DM twice (called instance A and B) for database MyDBase. If you change data in instance A and save the changes, you need to run a SELECT statement in instance B to see those changes.

➤ To start SQL DM from PCC

1 Start PCC if it is not already running. (Click **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center.**)

2 Expand the tree in the namespace (click the plus signs) until the **Tables** node appears for the database in which you are interested.

3 In the PCC data pane, right-click on a table icon, then click **Open**. You may also double click the table icon.



SQL DM starts and runs a Select * From *table_name* statement, where *table_name* is the name of the table icon. The results are placed into the Grid Results tab.

Note SQL DM opens a separate instance for each table. This means that you can open multiple instances of SQL DM from PCC for the *same* table. Changes saved in one instance are reflected in another instance only if you run a query (or queries) in the other instance. For example, suppose you open SQL DM twice (called instance A and B) for table MyTable. If you change data in the grid in instance A and save the changes, you need to run a SELECT statement in instance B to see those changes.

➤ To identify machine and database currently open

1 Observe the SQL DM title bar.

The title bar shows the name of the open database and the name of the machine on which the database resides.

For example, the following image shows that DEMODATA is the open database and DOCLAB2 is the machine on which DEMODATA resides.

If the title bar does **not** show the name of a database and the name of a machine, then no database is open.

➤ To open a database

1 Click **File** ▸ **Open Database**.

The **Login to database** dialog appears.

If the desired database has Pervasive.SQL security enabled, type the user name and password required for the database in the **User** and **Password** fields, respectively. By default, the user name is Master. If the database does not have security enabled, you may leave the **User** and **Password** fields blank.

2 Click a desired server in the **Server** list.

3 Click a desired database in the **Database** list.

4 Click **OK**.

➤ To create an SQL query or script

By default, when you start SQL DM, Query Pane is ready for you type in SQL statements. Statements that you build with Query Builder also appear in Query Pane. Note that Query Builder clears the contents of Query Pane, so you will lose any statements typed in unless you have saved them as a script. A script is one or more SQL statements saved as an ASCII file. See "To save an SQL query as a script" on page 6-47.

1 Perform one of the following actions:

a. Type the SQL query into Query Pane.

Separate SQL statements with a delimiter. By default, SQL DM uses the pound sign (#). You may change this delimiter character to the semi-colon (;). See "To set SQL statement separator" on page 6-51.

Note that you must use the pound sign as the statement delimiter in Query Pane if you create stored procedures. Stored procedures use the semicolon as a statement delimiter within the procedure itself.

b. Click **File ▸ New SQL Query** or click .

If Query Pane contains statements that have not been saved, you are prompted to save them. Click **Yes** to save the existing statements, specify a file name and location, then click **Save**. Click **No** to discard the existing statements.

Any SQL statements in Query Pane are cleared and the statement indicator is positioned at the top left of the pane. The indicator is the yellow arrow .

c. Use Query Builder to build a statement. See "Query Builder Tasks" on page 6-41.

If execution stops because of an error, the statement indicator positions to the statement that was being run when the interruption occurred. Also, the Text Results tab lists the statement that was last run. Knowing the last statement run can help you troubleshoot the problem. See "To continue running SQL statements that were stopped" on page 6-57.

➤ To close a database

1 Click **File ▸ Close Database**.

2 Click **Yes** to confirm that you want to close the database.

Note that the title bar no longer shows the name of an open database or the name of a machine on which the database resides. In addition, the commands and icons for running a query become inactive (grayed out). You can run a query only if a database is open.

➤ To change data with Grid Results grid

The Grid Results grid must be active before you can change values in cells. See "To set Grid Results to allow changes to database data" on page 6-58.

Note If you have a table without primary keys that contains records with duplicate data for a column, you will be unable to update the duplicate data on the Grid Results tab. SQL DM cannot tell one record from another because of the duplicate data. This problem does not occur if your table contains primary keys because they prevent duplicate data. (You may use an UPDATE statement to change duplicate data.)

1. Click the **Grid Results** tab.
2. Populate the grid by running a SELECT statement in Query Pane. Click **Tools ▸ Query ▸ Run Current SQL statement in grid** or click .

   After the grid has been populated by a SELECT statement, you may directly update the table by making changes to the values in the active grid. You must have the proper table permissions to affect the database data.
3. Change the data in one or more grid cells for a particular row.

   An icon that indicates a changed value appears in the leftmost column of the row:
4. Perform an action in SQL DM that causes the changed data to be written to the database. The following actions result in the data being changed:
   - Clicking the leftmost column of the row containing the changed data (clicking the icon for the row)
   - Moving to a grid row or a cell different from the one containing the changed data (you may move with the Tab key, arrow keys, or by using the mouse)
   - Running a SELECT statement
   - Closing the database (**File ▸ Close Database**)
   - Exiting SQL DM (**File ▸ Exit**).

Note You may see the row in which you change a value move to a different location within the grid. An example helps clarify why this occurs.

Suppose you have a table, t1, with a single integer column, col1, that contains the values 1 through 5 in ascending order. If you run a SELECT * statement against t1, SQL DM returns 5 records. Now assume you delete record 3 (DELETE FROM t1 WHERE col1=3). Your table now has a "hole" in it where record 3 existed. If you run a SELECT * statement again, SQL DM returns 4 records (1, 2, 4, and 5).

Now assume you add another value for col1 at the *bottom* of the Grid Results tab (you type in "6" at the bottom of the integer column and press the Tab key). The record containing the value "6" moves in the grid to the 3rd position from the top. In other words, the new record occupies the "hole" that resulted from the deletion of the other record.

This is the normal behavior of the Grid Results tab. The SRDE fills any "holes" in a table that have resulted from deleting records. SQL DM just queries the database and displays the natural ordering of the records. Once all of the "holes" are filled, records are added to the end of the existing records.

➢ To save an SQL query as a script

1 Perform one of the following actions:

a. Click **File ▸ Save SQL Query** or click .

If you do not have an SQL script already open, a Save As dialog appears. SQL DM assigns a default name to the query. The default name is query*n*.sql, where *n* is an integer that increments by one. Query1.sql is the first default name, query2.sql the second, and so forth.

If you have an SQL script already open, a Save As dialog does not appear. SQL DM saves the SQL statements in Query Pane with the name of the open script, even if all lines from the open script have been cleared and new lines typed. For example, suppose you open MySQLScript.sql then click **File ▸ New SQL Query**. Query Pane is cleared and the statement indicator is positioned to the top of the pane. You type an SQL statement then click **File ▸ Save SQL Query**. The SQL statement is saved as file MySQLScript.sql.

b. Click **File ▸ Save SQL Query As**.

A Save As dialog appears. If you have an SQL script open, this option lets you save the query with a different file name.

➢ To open an SQL script

1 Click **File ▸ Open SQL Query** or click .

If Query Pane contains statements that have not been saved, you are prompted to save them. Click **Yes** to save the existing statements, specify a file name and location, then click **Save**. Click **No** to discard the existing statements.

2 Type in the file name of the query you want to open in the **File name** field, or navigate to the location of the file and click on the file.

An SQL query file is an ASCII file of one or more SQL statements. By default, an SQL query files saved from SQL DM has a file extension of "sql."

3 Click **Open**.

SQL DM places the contents of the open script into Query Pane. Note that if you change the script then click **File ▸ Save SQL Query** or click , the changes overwrite the script file. If you want to save the changes to a different location or as a different script name, click **File ▸ Save SQL Query As.**

➢ To display help

1 Click **Help**, then the type of help you want:

- **How Do I?** – Lists the tasks that you can perform with SQL DM..
- **Topics** – Lists the documentation topics pertaining to SQL DM.
- **SQL Reference** – Displays an alphabetical list of SQL statements and keywords that are links to the SQL Engine Reference. (Statements not supported by SQL DM are listed below the alphabetical list.)
- **Search Documentation** – Searches for a string throughout the installed set of Pervasive.SQL documentation.
- **What's This?** – Identifies objects in the SQL DM GUI.
- **Glossary** – Defines terms used in the Pervasive.SQL documentation and products.

- **Online Manuals (Web)** – Directs your browser to www.pervasive.com where you can reference the online manuals.
- **Knowledge Base (Web)** – Directs your browser to www.pervasive.com where you can query the Pervasive.SQL Knowledge Base for additional information about Pervasive.SQL.
- **Developer Center (Web)** – Directs your browser to www.pervasive.com where you can research information specifically for application developers using Pervasive.SQL.
- **About** – Displays the name and version of SQL DM and the Pervasive.SQL copyright information.

➢ To exit SQL DM

1 Click **File ▸ Exit**. (You may also click the "X" close button in the upper right corner of the GUI.)

If Query Pane contains statements that have not been saved, you are prompted to save them. Click **Yes** to save the existing statements, specify a file name and location, then click **Save**. Click **No** to discard the existing statements.

Query Pane Tasks

➢ To show or hide Query Pane

1 Perform one of the following actions:

a. Click **View ▸ Query Pane**.

b. Click .

These actions function as a toggle. That is, if Query Pane is hidden, the action shows it, and vice versa. A database must be open to enable the command or icon.

➢ To move the cursor within Query Pane without running any SQL statements

The statement indicator, the yellow arrow in the left margin of Query Pane, indicates the statement in which the cursor is located.A statement delimiter is used to separate statements. See "To set SQL statement separator" on page 6-51.

1 Perform one of the following actions to move the cursor in Query Pane:

   a. Click on the statement you to which you want to move the cursor.

   b. Use the keyboard arrow keys to move the cursor to the desired statement.

➤ To move the cursor within Query Pane and run the SQL statement to which it moves

The statement indicator, the yellow arrow in the left margin of Query Pane, indicates the statement in which the cursor is located.A statement delimiter is used to separate statements. See "To set SQL statement separator" on page 6-51.

1 Perform one of the following actions to move the cursor in Query Pane:

   a. Click to move the cursor to the first statement and run that statement.

   b. Click to move the cursor to the previous statement and run that statement.

   c. Click to move the cursor to the next statement and run that statement.

   d. Click to move the cursor to the last statement and run that statement.

Note SQL DM shows the appropriate results tab as you move through the statements. Only SELECT statements send results to the Grid Results tab (you may also send SELECT results to the Text Results tab). All other statements send results to the Text Results tab.

For example, suppose you have the following lines in Query Pane:

```
SELECT * FROM Person#
SELECT * FROM Faculty#
DELETE FROM Person WHERE Person.ID=104321686#
SELECT * FROM Class#
```

Assume that the cursor is located on the first line and you click ⏩ to move to the next line, which contains a SELECT statement. The results show on the Grid Results tab. You click ⏩ again to move to the next line, which contains a DELETE statement. The results show on the Text Results tab. You click ⏩ again to move to the next line, which contains another SELECT statement. The results show on the Grid Results tab.

See also "Results Tabs" on page 6-4 and "To type a comment into Query Pane" on page 6-52.

➤ To set SQL statement separator

Two or more SQL statements in Query Pane must be separated by a delimiter. By default, SQL DM sets the delimiter to the pound sign (#). You may change the delimiter to the semicolon (;) if you choose.

Note You must use the pound sign as the statement delimiter in Query Pane if you create stored procedures. Stored procedures use the semicolon as a statement delimiter within the procedure itself.

1 Click **Tools ▸ Properties**.

2 Click the semicolon (;) in the list for **SQL statements separator**.

3 Click **OK**.

➤ To set a query time-out period

The query time-out period specifies how many seconds to allow an SQL statement to run before the SRDE cancels the statement. The default is zero seconds, which means that no time-out period is specified. The value you specify is stored in the registry and will also apply the next time you start SQL DM.

The time-out period applies to SELECT, UPDATE, INSERT, and DELETE statements, but not to Data Definition Language (DDL) statements (such as CREATE INDEX, ALTER TABLE, DROP TABLE, and so forth). The SRDE stops processing a query after the specified period, returns SQL_ERROR, and sets SQLSTATE to "S1T00." Because SQL statements are atomic, no records are updated, inserted, or deleted if the SRDE cancels an UPDATE, INSERT, or DELETE statement, respectively.

1 Click **Tools ▸ Properties**.

2 Type a value for **Query time-out**.

The allowed range is 0 through 4,294,967,295. Partial seconds (for example, 30.5) and negative numbers are not allowed.

3 Click **OK**.

➤ To wrap text within Query Pane

By default, word wrap is turned off, which is indicated by lack of a checkmark beside the command.

1 Perform one of the following actions:

a. Right-click within Query Pane then click **Word Wrap** (or press `Control+W`).

b. Click **View ▸ Word Wrap**.

Note that these actions function as a toggle (alternately activates and deactivates word wrap).

➤ To type a comment into Query Pane

Comments are indicated by double dashes (--). Each comment must be on a separate new line or after the statement separator on an existing line. SQL DM does **not** support the use of a start/end comment block (such as /* */). See "To set SQL statement separator" on page 6-51.

1 Click at the beginning of the line where you want a comment.

2 Type "--" followed by your comment text.

The following example shows valid comments.

```
SELECT * FROM t1#
-- This is a valid comment
-- and so is this
SELECT * FROM t2# -- This is valid after the # sign
```

The following example shows *invalid* comments.

```
SELECT * FROM t1 -- This is NOT valid before the #
/* This is NOT a valid comment
because "/*" and "*/" are not supported */
SELECT * FROM t2#
```

Note If the cursor is positioned *after* the statement separator, the next statement is current. This is true even if the statement indicator, the yellow arrow , is still on the line containing the cursor.

An example helps clarify this. Suppose you have the following lines in Query Pane:

```
SELECT * FROM Person# -- This is a comment
SELECT * FROM Faculty#
SELECT * FROM Class#
```

You run the first statement, then position the cursor in front of the "t" in "This." The cursor is positioned after the statement separator (#) on that line. The statement indicator still points to the line SELECT * FROM Person, but the actual current line is SELECT * FROM Faculty. If you click to move to (and run) the next line, the statement indicator moves to (and runs) SELECT * FROM Class.

➤ To cancel (undo) typing actions in Query Pane

1 Perform one of the following actions:

   a. Click **Edit ▸ Undo** (or press `Control+Z`) for each typing action that you wish to cancel.

   For example, if you have typed "Select * From Person" and you click **Edit ▸ Undo**, the word "Person" is removed. If you click **Edit ▸ Undo** again, the word "From" is removed, and so forth.

   b. Right-click within Query Pane, then click **Undo** (or press `Control+Z`).

➤ To cancel an Undo action (redo) in Query Pane

The Redo action performs the opposite of the last Undo action. For example, if the last Undo action removed the word "Select" from a statement, Redo restores the word "Select" to the statement.

1 Right-click within Query Pane, then click **Redo** (or press `Control+Y`) for each Undo action that you want to cancel.

➤ To find text in Query Pane

1 Click **Edit ▸ Find** (or press `Control+F`).

   A dialog appears on which you specify a text string to find.

2 Type, or paste, a text string into the **Find what** field.

3 Specify any options for the search, such as case sensitivity and direction.

4 Click **Find Next**.

5 To repeat the search for the same string, click **Edit ▸ Find Next** (or press `F3`).

➤ To replace text in Query Pane

1 Click **Edit ▸ Replace** (or press `Control+H`).

A dialog appears in which you specify a text string to search for and a text string with which to replace it.

2 Type, or paste, a text string into the **Find what** field.

3 Type, or paste, a text string into the **Replace with** field.

4 Specify any options for the search, such as case sensitivity and direction.

5 Perform one of the following actions:

- **a.** Click **Find Next** to locate the next occurrence of the search string, then click **Replace** to change the search string to the replace string. Repeat these actions as desired.
- **b.** Click **Replace All** to change all occurrences of the search string to the replace string.

6 Click **Cancel** to exit the dialog.

➤ To select text in Query Pane

1 Perform one of the following actions:

- **a.** Click **Edit ▸ Select All**, right-click within Query Pane then click **Select All**, or press `Control+A`. Any of these actions selects all of the contents of Query Pane.
- **b.** Press and hold down the left mouse button and drag the cursor across the text you want to select.
- **c.** Press and hold down the **Shift** key, then press a cursor arrow key to move the cursor across the text you want to select.

➤ To copy text from Query Pane

1 Perform one of the following actions:

- **a.** Click **Edit ▸ Copy** (or press `Control+C`).

b. Right-click on the selected text then click **Copy**.

Copied text is placed in the clipboard.

➤ To paste within Query Pane

1 Perform one of the following actions:

a. Click **Edit ▸ Paste** (or press `Control+V`).

b. Right-click within Query Pane then click **Paste**.

Text from the clipboard is placed into Query Pane.

➤ To cut or delete text from Query Pane

Text cut from Query Pane is placed in the clipboard and is available for pasting. Text deleted from Query Pane is not placed in the clipboard.

1 Perform one of the following actions:

a. To cut text, click **Edit ▸ Cut**, right-click within Query Pane then click **Cut**, or press `Control+X`.

b. To delete text, press the **Delete** key.

➤ To print the contents of Query Pane

1 Click **File ▸ Print**.

A dialog appears on which you specify the options for printing.

2 Specify any options for printing, such as printer name, printer range, and so forth.

3 Click **OK**.

➤ To layout the printed page

1 Click **File ▸ Page Setup**.

A dialog appears on which you specify the layout for the printed page.

2 Specify any options for the printed page, such as paper size, margins, and so forth.

3 Click **OK**.

➤ To toggle insert/overtype mode

1 Press the Insert key to toggle the text entry mode between insert and overtype.

Insert mode adds characters between other characters as you type in Query Pane. Existing characters are pushed to the right. Overtype mode replaces characters as you type.

The default mode is insert.

When the Insert/Overtype indicator shows OVR in a dimmed typeface, SQL DM is in insert mode. When **OVR** is boldface, SQL DM is in overtype mode.

Insert Mode	Overtype Mode

➤ To set the font for Query Pane

Setting the font affects the font used in the Query Pane, the Grid Results tab, and the Text Results tab.

1 Click **View ▸ Set Font**.

2 Specify the desired font characteristics, such as style and size.

3 Click **OK**.

Execution and Results Tasks

➤ To run a single SQL statement

A database must be open and one or more SQL statements must exist in Query Pane before you can run a statement.

1 Move to the SQL statement you want to run. See "To move the cursor within Query Pane without running any SQL statements" on page 6-49.

2 Perform one of the following actions:

   a. Click **Tools ▸ Query ▸ Run Current SQL statement in grid** or click .

   b. Click **Tools ▸ Query ▸ Run Current SQL statement in text** or click .

Note SQL DM automatically uses the Text Results tab for the results of SQL statements that are not SELECT statements. Only SELECT statements use the Grid Results tab.

➤ To run all SQL statements

A database must be open and one or more SQL statements must exist in Query Pane before you can run all statements. Two or more SQL statements in Query Pane must be delimited. See "To set SQL statement separator" on page 6-51.

1 Click **Tools** ▸ **Query** ▸ **Run All SQL Statement** or click .

➤ To stop running SQL statements

1 Click **Tools** ▸ **Query** ▸ **Stop** or click .

The statement indicator positions to the statement that was being run when the interruption occurred. The indicator is the yellow arrow .

If you stop the running of SQL statements, or if execution stops because of an error, the Text Results tab lists the statement that was last run. Knowing the last statement run can help you troubleshoot problems.

➤ To continue running SQL statements that were stopped

1 Click **Tools** ▸ **Query** ▸ **Continue**.

Execution of the SQL statements continues with the statement that contains the statement indicator. Note that you may move to a statement after execution stops. For example, suppose execution stops at statement 50 in a script that contains 100 statements. You could click on statement 60 and continue execution from there.

Unless you move the statement indicator, execution continues with the next statement following the one that last ran.

➤ To set the font for grid results or text results

Setting the font affects the font used in the Query Pane, the Grid Results tab, and the Text Results tab.

1 Perform one of the following actions:

   a. Click **View ▸ Set Font**

   b. Right-click within the Grid Results tab or the Text Results tab then click **Font**.

2 Specify the desired font characteristics, such as style and size.

3 Click **OK**.

➤ To work with grid results

1 Click the **Grid Results** tab.

After the grid has been populated by a SELECT statement, you may directly update your database data by changing the values in an active grid. You must have the proper table permissions to affect the database data. The grid must be active before you can change values in cells. See "To set Grid Results to allow changes to database data" on page 6-58.

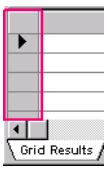
➤ To set Grid Results to allow changes to database data

Note A grid that allows changes to database data is referred to as "active." This means that if you change data in one of the grid cells, you change the data in your database provided you have the proper table permissions. By default, SQL DM starts with the Grid Results grid active.

See also "To set default behavior for Grid Results tab" on page 6-59.

1 Click **Tools ▸ Active Grid**.

A checkmark beside the **Active Grid** command indicates that the grid is active. In addition, an indicator column appears as the leftmost column in the grid:



The indicator column contains no column label.

➤ To set default behavior for Grid Results tab

Note A grid that allows changes to database data is referred to as "active." This means that if you change data in one of the grid cells, you change the data in your database provided you have the proper table permissions. By default, SQL DM starts with the Grid Results grid active.

See also "To set default behavior for Grid Results tab" on page 6-59.

1 Click **Tools ▸ Properties**.

2 Click the option **Active grid mode on startup** to set or clear the default behavior.

A checkmark beside the option indicates that the grid will be active each time SQL DM is started.

3 Click **OK**.

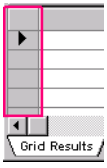
➤ To delete a row(s) from Grid Results tab

Note that the delete action has no undo. The grid must be active to allow deletion of rows. See "To set Grid Results to allow changes to database data" on page 6-58.

1 Right-click on the row (the record) that you want to delete, then click **Delete Row(s)**.

You may also select and delete multiple rows. To select multiple rows, press and hold down the **Shift** or **Control** key, then click the indicator column for the desired rows. Once the desired rows are selected, right-click and click **Delete Row(s)**.

The indicator column is the leftmost column in the grid:



Once a row(s) is selected, you may also press the **Delete** key to delete the row(s).

Note that you cannot delete a row that contains invalid values in one or more grid columns. For example, assume that you want to delete a record from the "Person" table provided with the sample database DEMODATA.

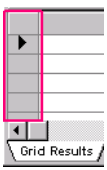
You run a `SELECT * FROM person` statement in Query Pane. You click on an ID cell in the grid and remove the ID value with the Backspace key. You then decide that you want to delete the entire record. The Delete Row(s) command does not work because the ID column contains an invalid value (it is blank). The ID column in the "Person" table does not allow NULLs. You would first need to add a valid value into the ID cell, then delete the entire record.

➤ To copy data from Grid Results tab to clipboard

Copied data is placed in the clipboard in a tab delimited format.

1 Perform one of the following actions to select the desired data:

a. Click the indicator column for the row(s) that you want to copy. The indicator column is the leftmost column in the grid:



To select multiple rows, press and hold down the left mouse button. Move the mouse pointer across the column indicator(s) for the row(s) you wish to select. You may also press and hold down the **Shift** or **Control** key, then click the indicator column for the desired row(s).

b. Right-click on a cell (to set a starting place), then press and hold down the left mouse button. Move the mouse pointer across the cell(s) you wish to select.

c. Left-click within a cell, then press and hold down the **Shift** key. Use the cursor arrow keys to extend the selection.

d. Right-click on a single cell to select the contents of only that cell. Ensure that no other rows or cells are selected.

2 Right-click, then click **Copy**.

The copied data can be pasted into another application, such as Microsoft Excel. For example, in Excel, you would click on a cell, right-click, then click Paste.

➤ To work with text results

1 Click the **Text Results** tab.

The Text Results tab shows in a text format the result of running SQL statements. You cannot change the data values in the database by changing the text.

➤ To clear results from Text Results tab

1 Right-click within the Text Results tab then click **Clear**.

➤ To select text on Text Results tab

1 Perform one of the following actions:

a. Right-click within the Text Results tab then click **Select All**, or press `Control+A`. Either action selects all of the contents on the tab.

b. Press and hold down the left mouse button and drag the cursor across the text you want to select.

c. Press and hold down the **Shift** key, then press the cursor arrow keys to move the cursor across the text you want to select.

➤ To copy text from Text Results tab

1 Perform one of the following actions:

a. Press `Control+C`.

b. Right-click on the selected text then click **Copy**.

Copied text is placed in the clipboard.

➤ To cut or delete text from Text Results tab

Text must first be selected before you can cut or delete it. Text cut is placed in the clipboard and is available for pasting. Text deleted is not placed in the clipboard.

1 Perform one of the following actions:

a. To cut text, right-click within the Text Results tab then click **Cut**, or press `Control+X`.

b. To delete text, press the **Delete** key.

Query Builder Tasks

➤ To show or hide Query Builder Diagram

A database must be open before you can show or hide Query Builder Diagram.

1 Click **View** ▸ **Query Builder Diagram** or click .

These actions function as a toggle. That is, if Query Builder Diagram is hidden, the action shows it, and vice versa. See also "Behavior When Query Builder Diagram Contains No Tables" on page 6-12.

➤ To show or hide Query Builder Grid

A database must be open before you can show or hide Query Builder Grid.

1 Perform one of the following actions:

a. Click **View** ▸ **Query Builder Grid**.

b. Click .

These actions function as a toggle. That is, if Query Builder Grid is hidden, the action shows it, and vice versa. See also "Behavior of Query Builder Grid When Query Builder Diagram Contains No Tables" on page 6-15.

➤ To set query type for Query Builder

This task requires that Query Builder Diagram or Query Builder Grid be active (showing).

1 Click **Tools** ▸ **Query Builder** ▸ **Change Query Type** or click .

A dialog appears on which you may choose the type of query.

2 Perform one of the following actions:

a. Click **Select** to build a SELECT statement.

b. Click **Update** to build an UPDATE statement.

c. Click **Insert** to build an INSERT statement.

d. Click **Delete** to build a DELETE statement.

3 Click **OK**.

➤ To build a SELECT statement with Query Builder

This task serves as a mini-tutorial on how to build an SQL statement with Query Builder. This task addresses building a SELECT statement, but many of the actions also apply to building an UPDATE, INSERT, or a DELETE statement. For this reason, we suggest that you complete this task before attempting to build one of the other statement types.

This task presents the steps to accomplish the following:

- Obtain the ID number, first name, and last name of all persons in your database with a first name of "James" or "Lisa" and whose last name begins with an "N," "O," "P," "Q," "R," "S," or "T." Once obtained, the records need to be sorted by ascending ID number.
- Determine from your database the average amount owed by the students.
- Find which persons in your database are also on the faculty, and determine the department in which the faculty member teaches and the salary of the faculty member.

The database used for this task is the sample database provided with Pervasive.SQL, named DEMODATA. DEMODATA is located, by default, in the folder PVSW\demodata.

This task assumes the following:

- You have reviewed these sections in this chapter: "SQL Data Manager Concepts" on page 6-2 and "SQL Data Manager GUI Visual Reference" on page 6-16.
- SQL DM is running (see "To start SQL DM as a stand-alone application" on page 6-42 or "To start SQL DM from PCC" on page 6-43)
- The database DEMODATA is open (see "To open a database" on page 6-44)
- Both Query Builder Diagram and Query Builder Grid are hidden (see "To show or hide Query Builder Diagram" on page 6-62 and "To show or hide Query Builder Grid" on page 6-62)
- No tables are open in Query Builder Diagram (see "To remove a table from Query Builder Diagram" on page 6-87).

Your first goal is to obtain the ID number, first name, and last name of all persons in your database with a first name of "James" or "Lisa"

and whose last name begins with an “N,” “O,” “P,” “Q,” “R,” “S,” or “T.” Once you have the correct records, you need to sort them by ascending ID number.

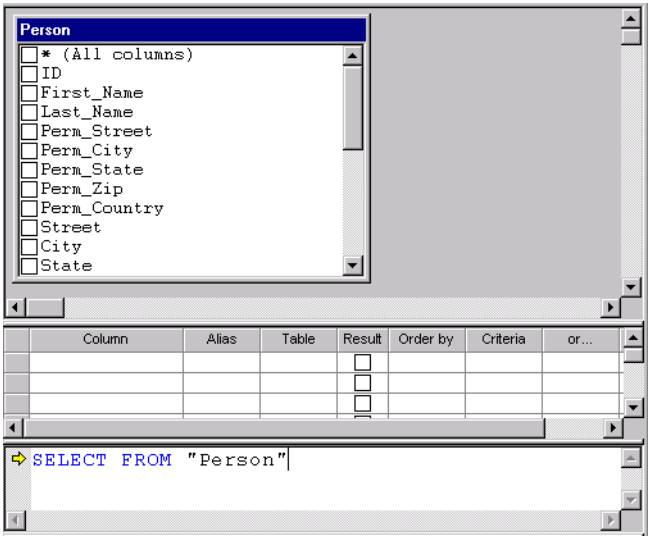
1 Click **View ▸ Query Builder Diagram** or click .

The Add Table dialog, the Query Builder Diagram, and the Query Builder Grid appear.

2 Click “Person” in the tables list then click **Add**.

3 Click **Close**.

The SQL statement is started in Query Pane as soon as you add a table to Query Builder Diagram. SQL DM should show something similar to the following:

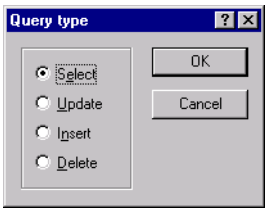


If you see a different type of grid, then you need to set Query Builder to build a SELECT statement. The next two steps show you how. Complete both steps even if you see the same type of grid.

4 Click **Tools ▸ Query Builder ▸ Change Query Type** or click .

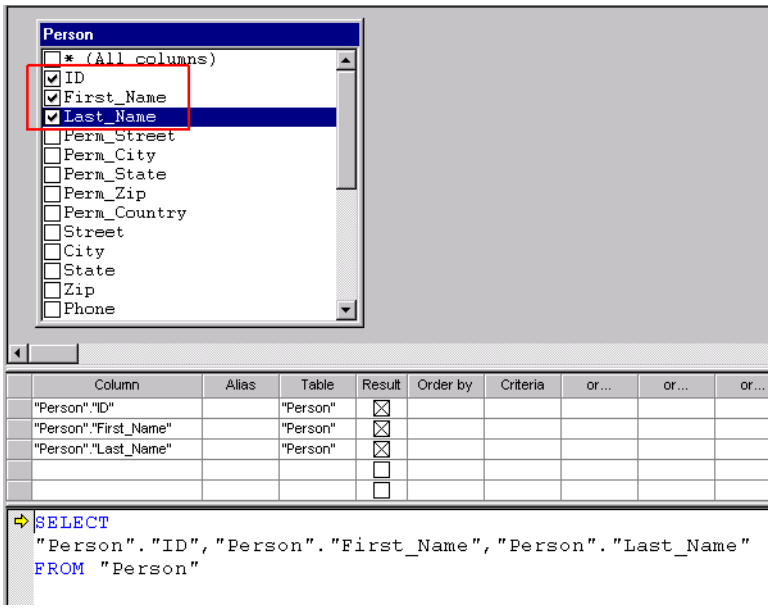
The “Query type” dialog appears.

5 Click **Select** then **OK**.



Query Builder is now set to build a SELECT statement.

6 In Query Builder Diagram, click the columns **ID**, **First_Name**, and **Last_Name** for table Person.

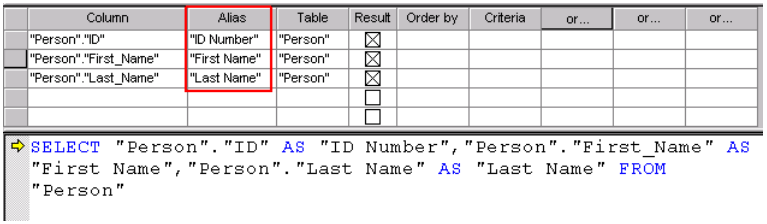


Notice that the columns appear in the grid and that the SQL statement in Query Pane continues to be built.

7 In the Alias column for "Person"."ID," "Person"."First_Name," and "Person"."Last_Name," type the following, respectively:

- **ID Number**
- **First Name**

- **Last Name**



Column	Alias	Table	Result	Order by	Criteria	or...	or...	or...
"Person"."ID"	"ID Number"	"Person"	☒					
"Person"."First_Name"	"First Name"	"Person"	☒					
"Person"."Last_Name"	"Last Name"	"Person"	☒					
			☐					
			☐					

```
SELECT "Person"."ID" AS "ID Number","Person"."First_Name" AS
"First Name","Person"."Last Name" AS "Last Name" FROM
"Person"
```

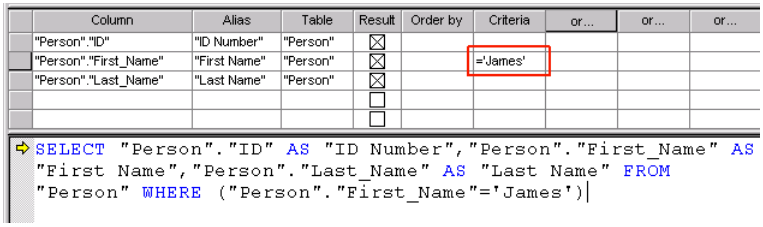
Notice that, as you move the cursor from each cell, the AS clauses are added to the statement and that double quotes are automatically placed around what you typed.

Note You must indicate to Query Pane that you are finished typing into a grid cell. You do this by moving the cursor from the cell once you have finished typing. For example, after you type "Last Name," move the cursor from that cell with the Tab key, cursor arrows, or mouse.

8 In the Criteria column for "Person"."First_Name," type **James**.

9 Press the **Tab** key.

Notice that Query Builder automatically adds an equals sign and single quotes around 'James' because "James" is a string data type. Non-string data types are not enclosed by quotes.

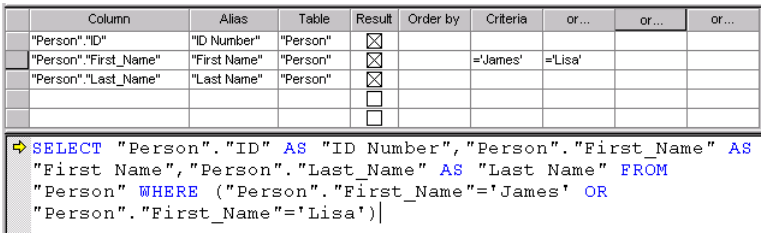


Column	Alias	Table	Result	Order by	Criteria	or...	or...	or...
"Person"."ID"	"ID Number"	"Person"	☒					
"Person"."First_Name"	"First Name"	"Person"	☒		='James'			
"Person"."Last_Name"	"Last Name"	"Person"	☒					
			☐					
			☐					

```
SELECT "Person"."ID" AS "ID Number","Person"."First_Name" AS
"First Name","Person"."Last_Name" AS "Last Name" FROM
"Person" WHERE ("Person"."First_Name"='James')
```

10 In the Or cell beside 'James,' type **Lisa** and press the **Tab** key.

SQL DM should show something similar to the following:



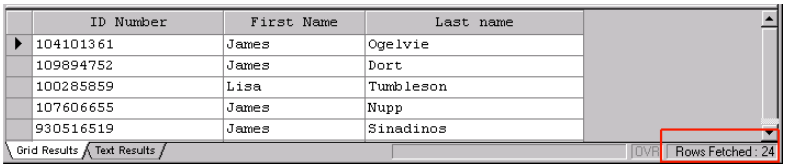
Column	Alias	Table	Result	Order by	Criteria	or...	or...	or...
"Person"."ID"	"ID Number"	"Person"	☒					
"Person"."First_Name"	"First Name"	"Person"	☒		='James'	='Lisa'		
"Person"."Last_Name"	"Last Name"	"Person"	☒					
			☐					
			☐					

```
SELECT "Person"."ID" AS "ID Number","Person"."First_Name" AS
"First Name","Person"."Last_Name" AS "Last Name" FROM
"Person" WHERE ("Person"."First_Name"='James' OR
"Person"."First_Name"='Lisa')
```

At this point, run the SQL statement to see what results are returned.

11 Click **Tools ▸ Query ▸ Run Current SQL Statement in Grid** or click .

The statement returns 24 records where first name equals "James" or "Lisa." But your goal also requires the records where the last name begins with an "N," "O," "P," "Q," "R," "S," or "T." You obtain the last name information by using a logical AND.



ID Number	First Name	Last name
104101361	James	Ogelvie
109894752	James	Dort
100285859	Lisa	Tumbleson
107606655	James	Nupp
930516519	James	Sinadinos

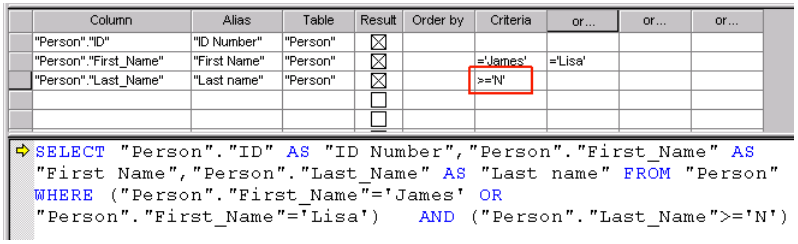
Grid Results | Text Results | OVR | Rows Fetched : 24

Also notice that the Aliases you specified appear as the column headings on the Grid Results tab.

Optional: If you want to view the results as text, click **Tools ▸ Query ▸ Run Current SQL Statement in Text** or click .

Click the **Grid Result** tab after you finish viewing the text results.

12 In the Criteria column for "Person"."Last_Name," type >=N (greater than or equal to N) and press the **Tab** key.

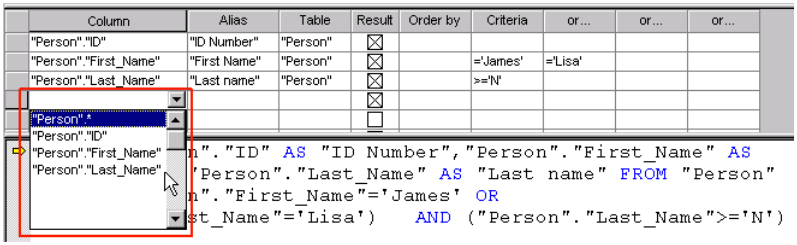


Column	Alias	Table	Result	Order by	Criteria	or...	or...	or...
"Person"."ID"	"ID Number"	"Person"	☒					
"Person"."First_Name"	"First Name"	"Person"	☒		='James'	='Lisa'		
"Person"."Last_Name"	"Last name"	"Person"	☒		>='N'			
			☐					
			☐					

```
SELECT "Person"."ID" AS "ID Number","Person"."First_Name" AS
"First Name","Person"."Last_Name" AS "Last name" FROM "Person"
WHERE ("Person"."First_Name"='James' OR
"Person"."First_Name"='Lisa')   AND ("Person"."Last_Name">='N')
```

Notice that the grid does not contain an AND column. You create a logical AND by adding the same database column to the grid and specifying the additional criteria.

13 Position the mouse pointer in the empty cell below "Person"."Last_Name" then press and hold down the left mouse button. A list of available database columns appears.

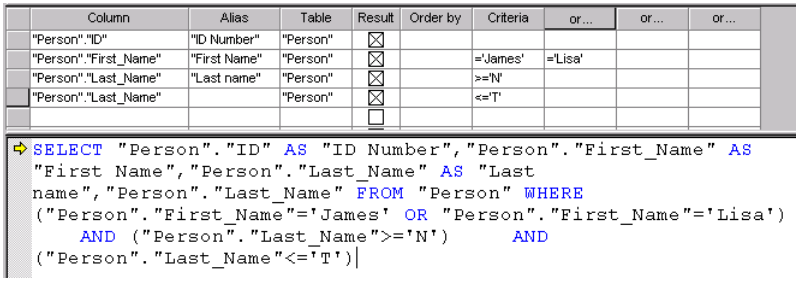


14 Click **"Person"."Last_Name"** in the list.

15 Press the **Tab** key 5 times to position the cursor to the Criteria column.

16 In the Criteria cell, type <=T and press the **Tab** key.

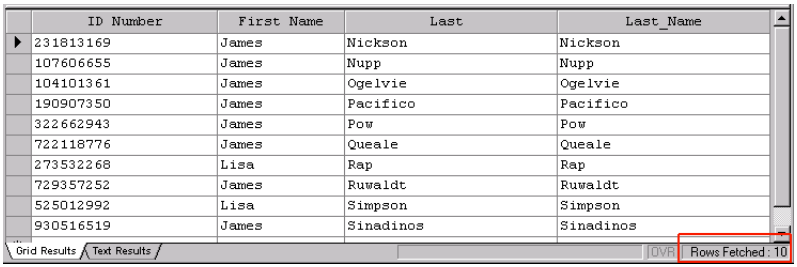
SQL DM should show something similar to the following:



Notice that the SQL statement now contains the AND clauses for last name.

17 Click **Tools** ▸ **Query** ▸ **Run Current SQL Statement in Grid** or click .

This time, the statement returns 10 records, but the records are not sorted. Their order is the sequence in which they were added to the database.

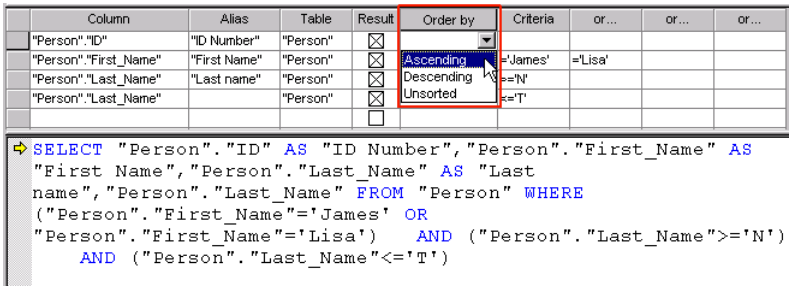


ID Number	First Name	Last	Last_Name
231813169	James	Nickson	Nickson
107606655	James	Nupp	Nupp
104101361	James	Ogelvie	Ogelvie
190907350	James	Pacifico	Pacifico
322662943	James	Pow	Pow
722118776	James	Queale	Queale
273532268	Lisa	Rap	Rap
729357252	James	Ruwaldt	Ruwaldt
525012992	Lisa	Simpson	Simpson
930516519	James	Sinadinos	Sinadinos

Grid Results | Text Results | OVR | Rows Fetched : 10

You are almost finished with your first goal! All you have remaining is to sort the records by the person ID numbers.

18 Position the mouse pointer in the **Order by** cell for "Person"."ID," then press and hold down the left mouse button.



Column	Alias	Table	Result	Order by	Criteria	or...	or...	or...
"Person"."ID"	"ID Number"	"Person"	☒					
"Person"."First_Name"	"First Name"	"Person"	☒	Ascending	='James'	='Lisa'		
"Person"."Last_Name"	"Last name"	"Person"	☒	Descending	>='N'			
"Person"."Last_Name"		"Person"	☒	Unsorted	<='T'			
			☐					

```
SELECT "Person"."ID" AS "ID Number","Person"."First_Name" AS
"First Name","Person"."Last_Name" AS "Last
name","Person"."Last_Name" FROM "Person" WHERE
("Person"."First_Name"='James' OR
"Person"."First_Name"='Lisa')   AND ("Person"."Last_Name">='N')
    AND ("Person"."Last_Name"<='T')
```

19 Click **Ascending** in the list then press the **Tab** key.

The ORDER by clause is added to the SQL statement. Notice that the ascending symbol is added to the Person table for the ID column.

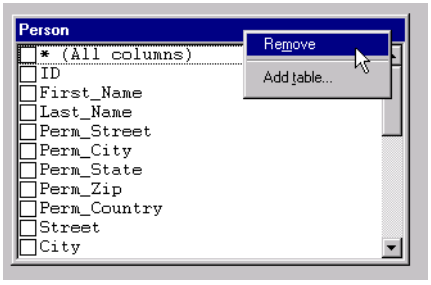
20 Click **Tools** ▸ **Query** ▸ **Run Current SQL Statement in Grid** or click .

The statement returns 10 records but now they are sorted in ascending sequence by ID number.

	ID Number	First Name	Last
▶	104101361	James	Ogelvie
	107606655	James	Nupp
	190907350	James	Pacifico
	231813169	James	Nickson
	273532268	Lisa	Rap
	322662943	James	Pow
	525012992	Lisa	Simpson
	722118776	James	Queale
	729357252	James	Ruwaldt
	930516519	James	Sinadinos

You have now completed your first goal for this task. Your next goal is to determine from your database the average amount owed by the students. You determine this with an aggregate query that uses a Group By clause. The "Person" table does not contain the data you need so you are finished with it.

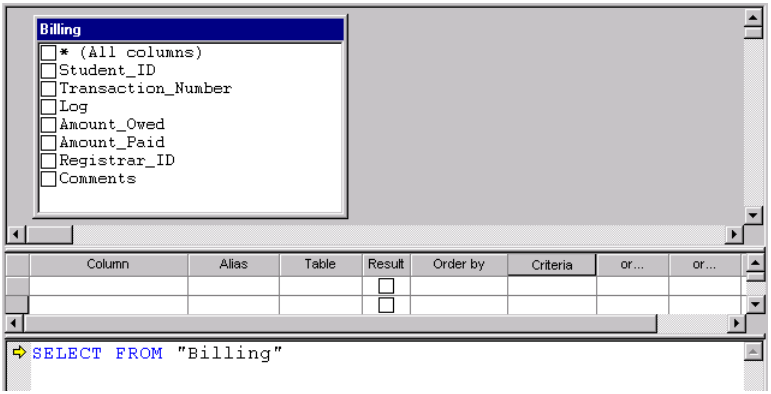
21 In Query Builder Diagram, right-click on the title bar of the "Person" table, then click **Remove**.



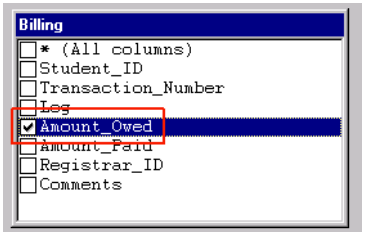
22 Right-click in the Query Builder Diagram, then click **Add table**.

23 On the Add Table dialog, click "Billing," then **Add**, then **Close**.

SQL DM should show something similar to the following:

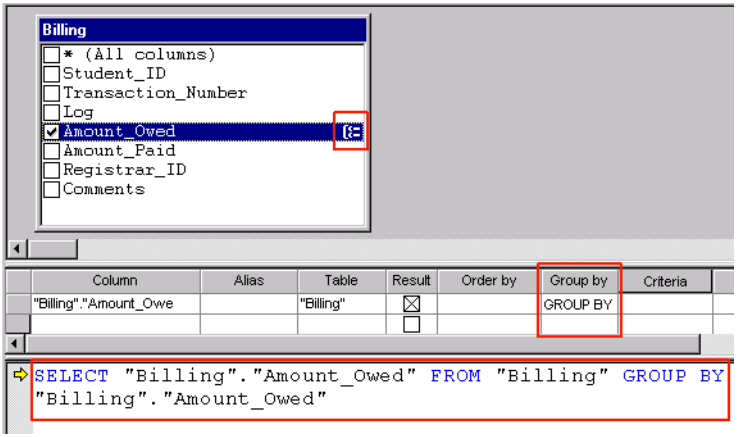


24 In Query Builder Diagram, click the column **Amount_Owed**.



25 Click **Tools** ▸ **Query Builder** ▸ **Group By** or click .

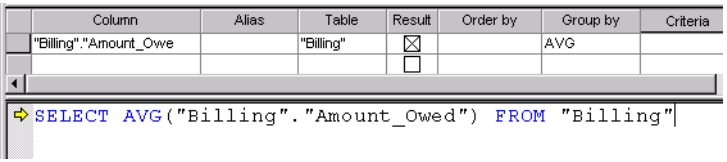
The Group By column appears on the grid, the Group By symbol is added to the "Billing" table for the Amount_Owed column, and the SQL statement now contains a Group By clause.



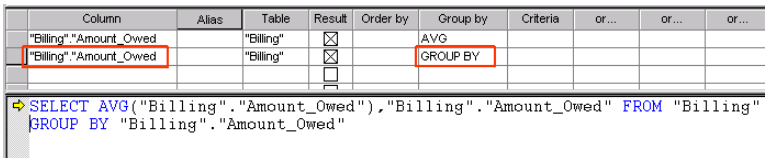
26 Position the mouse pointer in the **Group by** cell for "Billing"."Amount_Owed," then press and hold down the left mouse button.

27 Click **AVG** (for average) from the list then press the **Tab** key.

Query Pane now contains a query that determines the average amount owned by students.



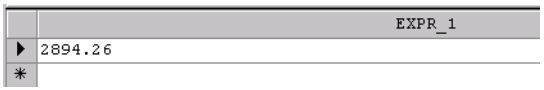
Notice that the Group By part of the clause went away when an aggregate function (AVG) was specified for the column. If you also want to group by the column, position the mouse pointer in the empty cell below "Billing"."Amount_Owed," then press and hold down the left mouse button. From the list, click "Billing"."Amount_Owed." The Group by column shows GROUP BY and the statement in Query Pane contains a GROUP BY clause.



In other words, you need one line in Query Builder Grid for each aggregate function or Group By clause that you want in the SQL statement.

28 Click **Tools** ▸ **Query** ▸ **Run Current SQL Statement in Grid** or click .

The average amount owed by students is $2,894.26.



Notice that the column heading for the output is EXPR_1. What if you wanted the heading to read "Average Amount Owed." What would you change on Query Builder Grid to produce this result? You would type "Average Amount Owed" as an Alias for "Billing"."Amount_Owed." Do that and run the statement again to see if the heading changes.

You have now completed your second goal for this task. Your last goal is to find which persons in your database are also on the faculty, and determine the department in which the faculty member teaches and the salary of the faculty member.

Because you now have some experience using Query Builder, the steps for this goal are more generic and contain fewer images of the GUI. If you are unsure how to complete a step, review the previous steps in this task.

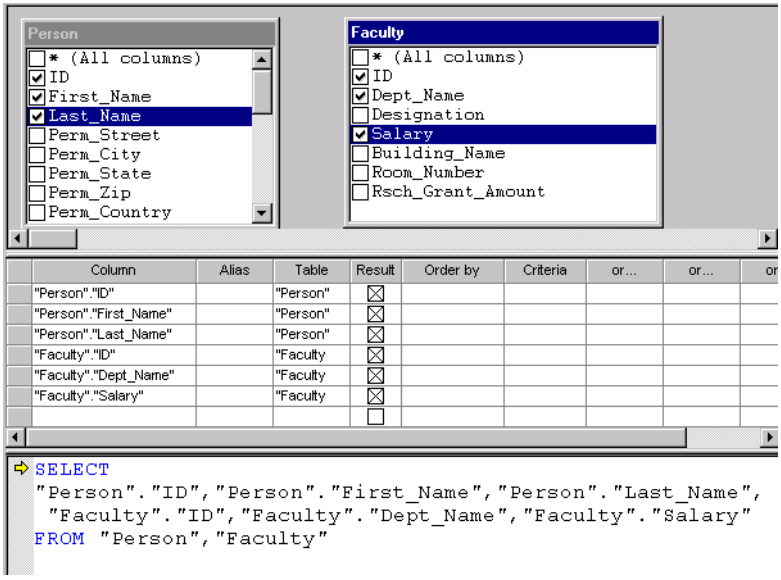
You are finished with the "Billing" table.

29 Remove the "Billing" table from Query Builder Diagram and add the tables "Person" and "Faculty."

30 In Query Builder Diagram, click the columns **ID**, **First_Name**, and **Last_Name** for table "Person."

31 In Query Builder Diagram, click the columns **ID**, **Dept_Name**, and **Salary** for table "Faculty."

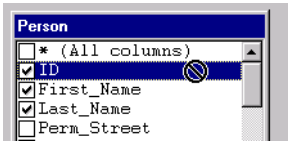
SQL DM should show something similar to the following:



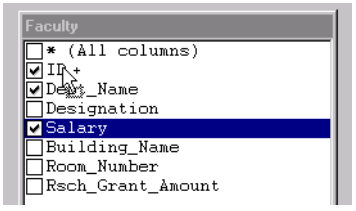
You are now ready to join the two tables with the ID column. The join allows you to determine which faculty members are also in the Person table. The following steps show you how to join the columns by dragging the column from one table to the corresponding column in the other column.

32 In Query Builder Diagram, click **ID** and keep the left mouse button held down.

The mouse pointer changes shape to ⃠:

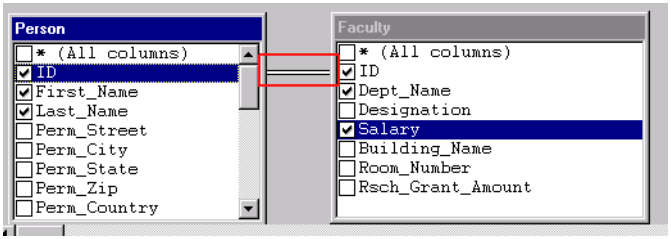


33 "Drag" the ID column to the Faculty ID column:



34 Release the mouse button.

A join line appears between the two tables:



Notice that the JOIN syntax is not used in the SQL statement. Instead, a WHERE clause is added:

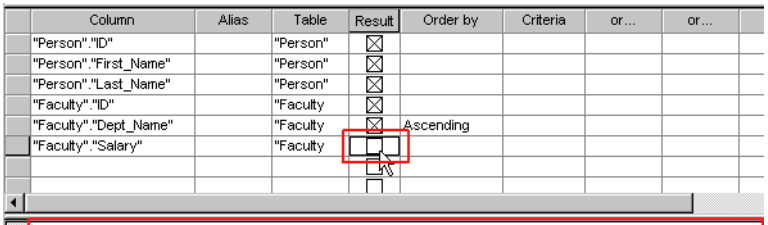
```
SELECT
"Person"."ID","Person"."First_Name","Person"."Last_Name",
"Faculty"."ID","Faculty"."Dept_Name","Faculty"."Salary"
FROM "Person","Faculty" WHERE ("Person"."ID" =
"Faculty"."ID")
```

35 Click **Tools** ▸ **Query** ▸ **Run Current SQL Statement in Grid** or click .

The statement returns 95 records. Your manager arrives and reviews the results. Your manager tells you to remove the salary information because it is not needed. How do you do this?

36 In Query Builder Grid, click on the **Result** option for "Faculty"."Salary."

This action removes the "X" from the option and also removes the Salary part from the selection list in the SQL statement.




```
SELECT
"Person"."ID","Person"."First_Name","Person"."Last_Name",
"Faculty"."ID","Faculty"."Dept_Name" FROM
"Person","Faculty" WHERE ("Person"."ID" =
"Faculty"."ID") ORDER BY "Faculty"."Dept_Name"
```

37 Run the query again.

You have now completed all goals for this task.

➤ To build an UPDATE statement with Query Builder

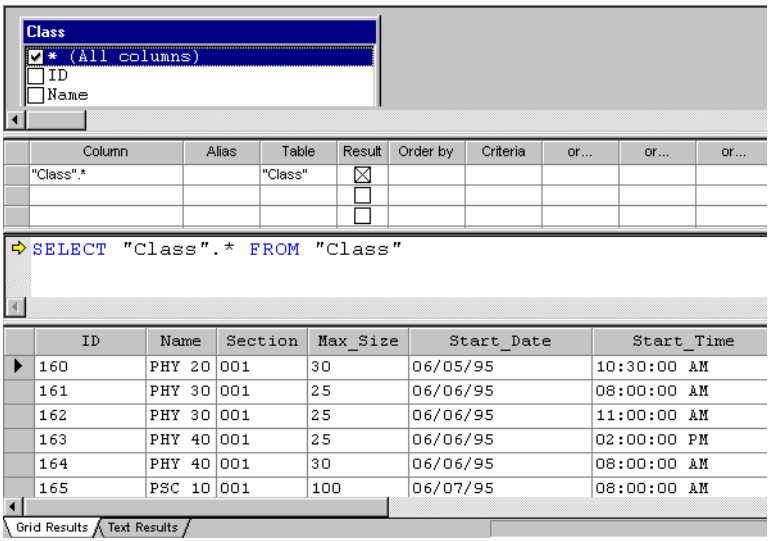
This task assumes the following:

- You have completed the task "To build a SELECT statement with Query Builder" on page 6-63. That tasks contains how-to information that is not covered again in this task.
- You have reviewed these sections in this chapter: "SQL Data Manager Concepts" on page 6-2 and "SQL Data Manager GUI Visual Reference" on page 6-16.
- SQL DM is running (see "To start SQL DM as a stand-alone application" on page 6-42 or "To start SQL DM from PCC" on page 6-43)
- The database DEMODATA is open (see "To open a database" on page 6-44)
- Both Query Builder Diagram and Query Builder Grid are hidden (see "To show or hide Query Builder Diagram" on page 6-62 and "To show or hide Query Builder Grid" on page 6-62)
- No tables are open in Query Builder Diagram (see "To remove a table from Query Builder Diagram" on page 6-87).

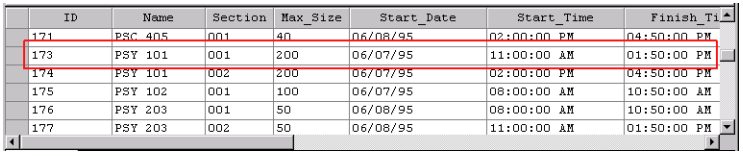
Your goal for this task is to change the time at which Psychology 101 begins to 10:30 am and reduce its maximum class size to 175. Before you make the changes, however, look at the information for this class.

1. Set Query Builder to build a SELECT statement (change the query type to "Select.")
2. Add table "Class" to Query Builder Diagram.
3. Click the column * (**All columns**) then run the SQL statement for output to the Grid Results tab.

SQL DM should show something similar to the following:



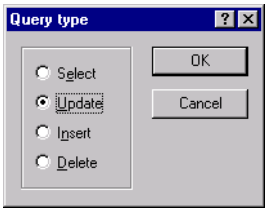
4 Scroll the results grid until you see the record for Psychology 101 that has a start time of 11:00 am.



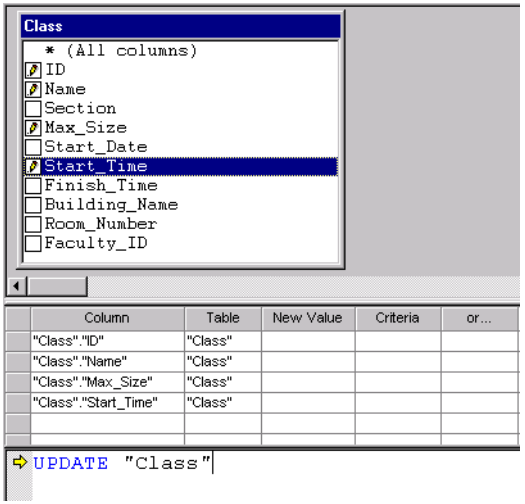
This is the record you want to change. Notice that its ID is 173.

5 Activate word wrap in Query Pane if it is not already activated.

6 Set Query Builder to build an UPDATE statement:



7 In Query Builder Diagram, click the columns **ID**, **Name**, **Max_Size**, and **Start_Time**.



Notice that the symbol appears in the table window instead of the ✔ symbol that appears for a SELECT statement.

8 In the Criteria column for "Class"."ID," type **173** and press the **Tab** key.

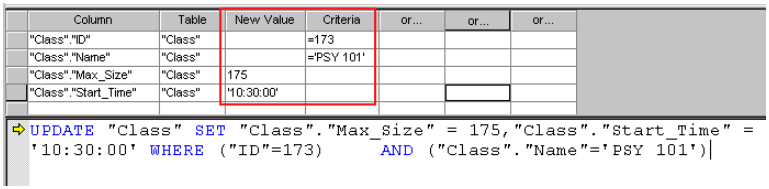
9 In the Criteria column for "Class"."Name," type **PSY 101** and press the **Tab** key.

You could build your statement without the class name because the class ID uniquely identifies the record. For clarity to someone reading the SQL statement, though, include the class name.

10 In the New Value column for "Class"."Max_Size," type **175** and press the **Tab** key.

11 In the New Value column for "Class"."Start_Time," type **'10:30:00'** and press the **Tab** key (remember to use single quotes around the time).

SQL DM should show something similar to the following:



Column	Table	New Value	Criteria	or...	or...	or...
"Class"."ID"	"Class"		=173			
"Class"."Name"	"Class"		='PSY 101'			
"Class"."Max_Size"	"Class"	175				
"Class"."Start_Time"	"Class"	'10:30:00'				

```
UPDATE "Class" SET "Class"."Max_Size" = 175,"Class"."Start_Time" =
'10:30:00' WHERE ("ID"=173)     AND ("Class"."Name"='PSY 101')
```

12 Click **Tools** ▸ **Query** ▸ **Run Current SQL Statement in Text** or click .

The Text Results tab contains the following result:

```
SQL statement (script) has executed successfully.
1 row(s) were affected.
```

Now look at the data in the "Class" table again to verify that you changed the record.

13 Set Query Builder to build a SELECT statement (change the query type to "Select.")

14 Click the column * (**All columns**) then run the SQL statement for output to the Grid Results tab.

15 Scroll the results grid until you see the record for Psychology 101 with ID 173.

Notice that the maximum class size is now 175 and the start time is 10:30 am.

ID	Name	Section	Max_Size	Start_Date	Start_Time
171	PSC 405	001	40	06/08/95	02:00:00 PM
173	PSY 101	001	175	06/07/95	10:30:00 AM
174	PSY 101	002	200	06/07/95	02:00:00 PM
175	PSY 102	001	100	06/07/95	08:00:00 AM
176	PSY 203	001	50	06/08/95	08:00:00 AM
177	PSY 203	002	50	06/08/95	11:00:00 AM

➤ To build an INSERT statement with Query Builder

This task assumes the following:

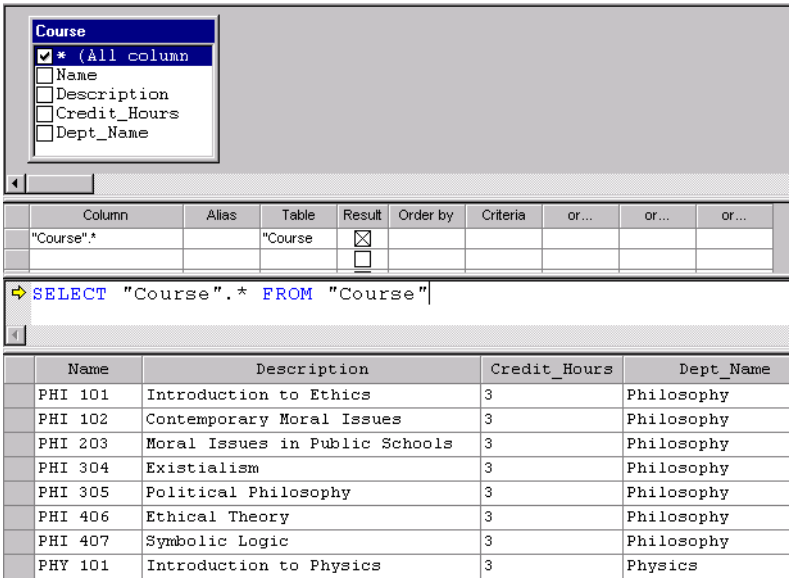
- You have completed the task "To build a SELECT statement with Query Builder" on page 6-63. That tasks contains how-to information that is not covered again in this task.
- You have reviewed these sections in this chapter: "SQL Data Manager Concepts" on page 6-2 and "SQL Data Manager GUI Visual Reference" on page 6-16.

- SQL DM is running (see “To start SQL DM as a stand-alone application” on page 6-42 or “To start SQL DM from PCC” on page 6-43)
- The database DEMODATA is open (see “To open a database” on page 6-44)
- Both Query Builder Diagram and Query Builder Grid are hidden (see “To show or hide Query Builder Diagram” on page 6-62 and “To show or hide Query Builder Grid” on page 6-62)
- No tables are open in Query Builder Diagram (see “To remove a table from Query Builder Diagram” on page 6-87).

Your goal for this task is to add a new philosophy class to the database. Before you make the change, however, look at the information for the existing courses.

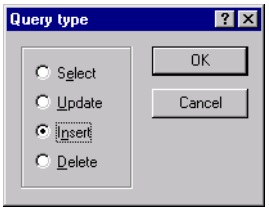
1. Set Query Builder to build a SELECT statement (change the query type to “Select.”)
2. Add table “Course” to Query Builder Diagram.
3. Click the column * (**All columns**) then run the SQL statement for output to the Grid Results tab.
4. Scroll the results grid until you see the records for Philosophy.

   SQL DM should show something similar to the following:

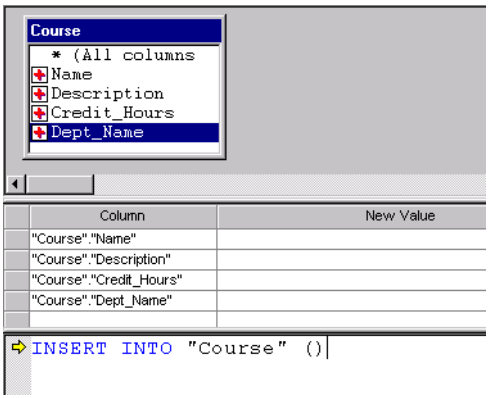


5 Activate word wrap in Query Pane if it is not already activated.

6 Set Query Builder to build an INSERT statement:



7 In Query Builder Diagram, click the columns **Name**, **Description**, **Credit_Hours**, and **Dept_Name**.



Notice that the ✚ symbol appears in the table window.

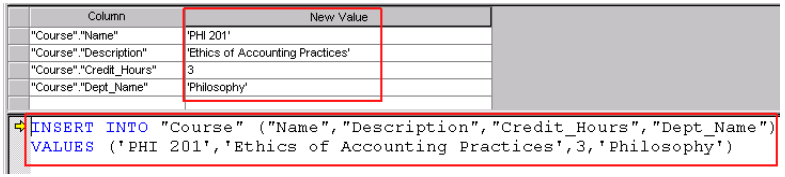
8 In the New Value column for "Course"."Name," type **PHI 201** and press the **Tab** key.

9 In the New Value column for "Course"."Description," type **Ethics of Accounting Practices** and press the **Tab** key.

10 In the New Value column for "Course"."Credit_Hours," type **3** and press the **Tab** key.

11 In the New Value column for "Course"."Dept_Name," type **Philosophy** and press the **Tab** key.

SQL DM should show something similar to the following:



Note that when you perform an INSERT, you must provide a value for all columns that cannot be null. For the Course table, "Course"."Name" and "Course"."Dept_Name" must contain values. For practice, however, this task has you provide values for all four columns.

12 Click **Tools ▸ Query ▸ Run Current SQL Statement in Text** or click .

The Text Results tab contains the following result:

```
SQL statement (script) has executed successfully.
1 row(s) were affected.
```

Now look at the data in the "Course" table again to verify that you inserted the record.

13 Set Query Builder to build a SELECT statement (change the query type to "Select.")

14 Click the column * (**All columns**) then run the SQL statement for output to the Grid Results tab.

15 Scroll the results grid until you find the record you inserted, PHI 201.

The philosophy class you inserted is now part of the table.

Name	Description	Credit_Hours	Dept_Name
SPA 408	Study of Spanish Literature	3	Spanish
THE 101	Theatre Performance	3	Theatre
THE 102	Theatre Laboratory	1	Theatre
THE 203	Makeup for Theatre	3	Theatre
THE 204	Introduction to Technical Theatre	3	Theatre
THE 305	Costume for Theatre	3	Theatre
THE 406	Advanced Production for Theatre	3	Theatre
PHI 201	Ethics of Accounting Practices	3	Philosophy

➤ To build a DELETE statement with Query Builder

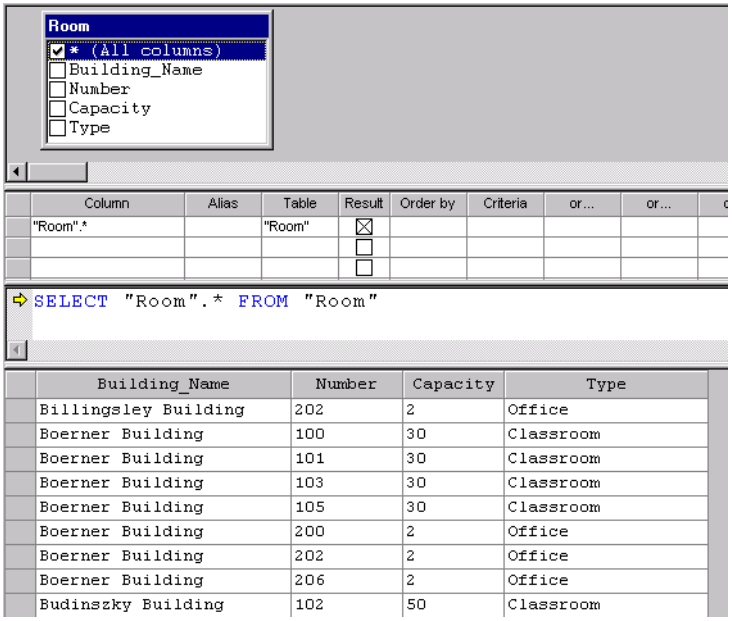
This task assumes the following:

- You have completed the task "To build a SELECT statement with Query Builder" on page 6-63. That tasks contains how-to information that is not covered again in this task.
- You have reviewed these sections in this chapter: "SQL Data Manager Concepts" on page 6-2 and "SQL Data Manager GUI Visual Reference" on page 6-16.
- SQL DM is running (see "To start SQL DM as a stand-alone application" on page 6-42 or "To start SQL DM from PCC" on page 6-43)
- The database DEMODATA is open (see "To open a database" on page 6-44)
- Both Query Builder Diagram and Query Builder Grid are hidden (see "To show or hide Query Builder Diagram" on page 6-62 and "To show or hide Query Builder Grid" on page 6-62)
- No tables are open in Query Builder Diagram (see "To remove a table from Query Builder Diagram" on page 6-87).

Your goal for this task is to remove from the database classroom 100 in the Boerner Building. Before you make the changes, however, look at the information for the existing classrooms.

1. Set Query Builder to build a SELECT statement (change the query type to "Select.")
2. Add table "Course" to Query Builder Diagram.
3. Click the column * (**All columns**) then run the SQL statement for output to the Grid Results tab.

4 Scroll the results grid until you see the records for the Boerner Building.

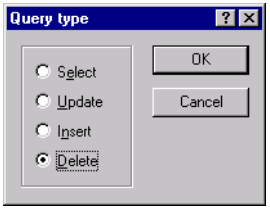


Building_Name	Number	Capacity	Type
Billingsley Building	202	2	Office
Boerner Building	100	30	Classroom
Boerner Building	101	30	Classroom
Boerner Building	103	30	Classroom
Boerner Building	105	30	Classroom
Boerner Building	200	2	Office
Boerner Building	202	2	Office
Boerner Building	206	2	Office
Budinszky Building	102	50	Classroom

Notice that room 100 is one of the rooms listed in the Boerner Building.

5 Activate word wrap in Query Pane if it is not already activated.

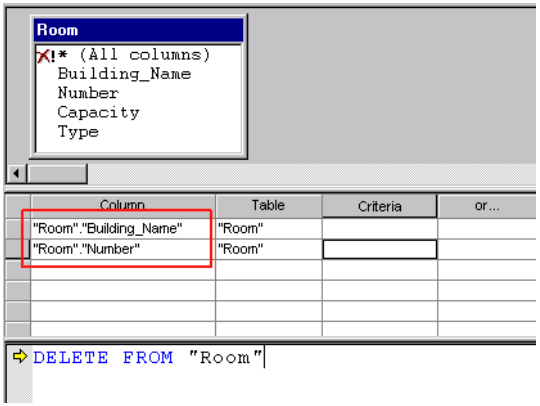
6 Set Query Builder to build a DELETE statement:



Notice in the table window that none of the columns are selectable for a DELETE statement.You cannot delete individual columns so check boxes do not appear in the table window. Instead of selecting columns, you specify the criteria in Query Builder Grid for the rows you want to delete, as explained next.

7 In the Query Builder Grid, specify that you want to work with columns "Room"."Building_Name" and "Room"."Name." (Remember to press the Tab key to move from the "Room"."Name" cell.)

SQL DM should show something similar to the following:

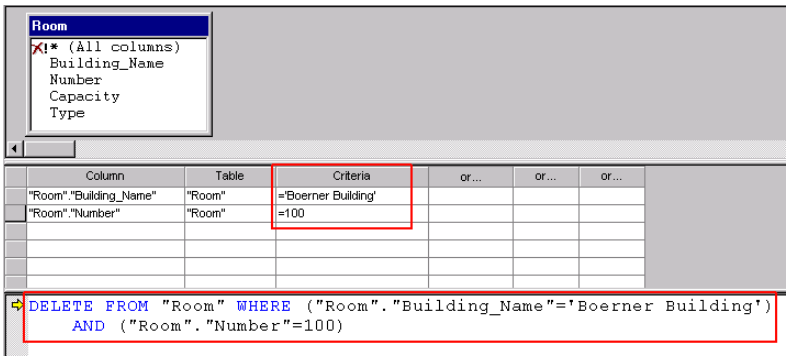


Why do you need both the building name and the room number? The "Room" table contains several records for the Boerner Building. If your only deletion criteria was building name, you would delete records you need to keep. Similarly, "Room" contains records for room 100 in other buildings, so you would not want to delete all rooms with room number 100.

8 In the Criteria column for "Room"."Building_Name," type **Boerner Building** and press the **Tab** key.

9 In the Criteria column for "Room"."Number," type **100** and press the **Tab** key.

SQL DM should show something similar to the following:



10 Click **Tools ▸ Query ▸ Run Current SQL Statement in Text** or click .

The Text Results tab contains the following result:

```
SQL statement (script) has executed successfully.
1 row(s) were affected.
```

Now look at the data in the "Room" table again to verify that you deleted the record.

11 Set Query Builder to build a SELECT statement (change the query type to "Select.")

12 Click the column * (**All columns**) then run the SQL statement for output to the Grid Results tab.

13 Scroll the results grid until you see the records for the Boerner Building.

Notice that room 100 is no longer listed as one of the classrooms for the Boerner Building.

	Building_Name	Number	Capacity	Type
	Billingsley Building	202	2	Office
	Boerner Building	101	30	Classroom
	Boerner Building	103	30	Classroom
	Boerner Building	105	30	Classroom
	Boerner Building	200	2	Office
	Boerner Building	202	2	Office
	Boerner Building	206	2	Office
	Budinszky Building	102	50	Classroom

➤ To add a table to Query Builder Diagram

This task requires that Query Builder Diagram be active (showing).

1 Perform one of the following actions:

a. Click **Tools ▸ Query Builder ▸ Add Table** or click .

b. Right-click in the pane of Query Builder Diagram, then click **Add table**.

A list appears that shows the tables in the active database.

2 Click the desired table in the list.

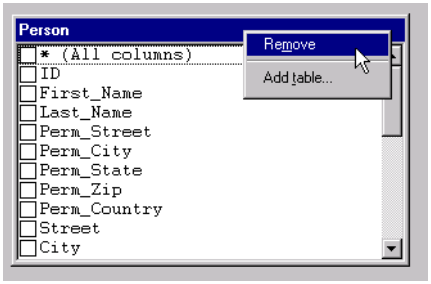
3 Click **Add**.

See also "Behavior When Query Builder Diagram Contains No Tables" on page 6-12.

➤ To remove a table from Query Builder Diagram

This task requires that Query Builder Diagram be active (showing).

1 Right-click on the title bar of a table in Query Builder Diagram.



2 Click **Remove**.

➤ To join tables in Query Builder Diagram

See steps 32 through 34 in "To build a SELECT statement with Query Builder" on page 6-63 for an example of joining tables.

➤ To insert a row into Query Builder Grid

This task requires that Query Builder Grid be active (showing).

1 Right-click on the row that you want to move *down*.

2 Click **Insert Row**.

The inserted row is placed *above* the row that contained the cursor.

➤ To clear the contents of a row(s) on Query Builder Grid

This task requires that Query Builder Grid be active (showing).

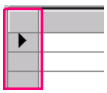
1 Right-click on the row that you want to clear.

2 Click **Clear Row(s)**.

The clear action removes the values from all of the grid cells for that row, but leaves the empty row in the grid. The SQL statement in Query Pane is rebuilt.

You may also select and clear multiple rows. To select multiple rows, press and hold down the **Shift** or **Control** key, then click the indicator column for the desired rows. Once the desired rows are selected, right-click and click **Clear Row(s)**.

The indicator column is the leftmost column in the grid:



➤ To delete a row(s) from Query Builder Grid

This task requires that Query Builder Grid be active (showing).

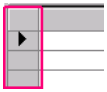
1 Right-click on the row that you want to delete.

2 Click **Delete Row(s)** (or press the **Delete** key).

The delete action removes the row in the grid. The SQL statement in Query Pane is rebuilt.

You may also select and delete multiple rows. To select multiple rows, press and hold down the **Shift** or **Control** key, then click the indicator column for the desired rows. Once the desired rows are selected, right-click and click **Delete Row(s)** (or press the **Delete** key).

The indicator column is the leftmost column in the grid:



➤ To add a Group By column to Query Builder Grid

See steps 25 through 27 in "To build a SELECT statement with Query Builder" on page 6-63 for an example of adding a Group By column.

chapter 7

Pervasive System Analyzer (PSA)

Usage Help for the Diagnostic Utility in Pervasive.SQL V8

Pervasive System Analyzer is a utility that performs many functions that include:

- Analysis and display of Pervasive.SQL components
- Archival, restoration, and deletion of Pervasive.SQL components
- Testing of network connectivity and of the functionality of the transactional and relational engines

The following are the sections found in this chapter:

- "PSA Concepts" on page 7-2
- "PSA GUI Visual Reference" on page 7-12
- "PSA Tasks" on page 7-15

PSA Concepts

The Pervasive System Analyzer (PSA) runs automatically as the first phase of installation in order to optimize your system for installation of the newest version: Pervasive.SQL V8. You may also run PSA at any time to analyze your system, archive Pervasive components, restore previously archived Pervasive components, delete Pervasive components and settings, or delete Pervasive archives.

PSA Functional Overview

PSA performs the following tasks:

- Analyzes your system to see if you have any previous versions of Pervasive.SQL installed and running.
- Archives previously installed Pervasive.SQL components.
- Restores components that you have previously archived using PSA.
- Deletes Pervasive.SQL components or archives.
- Displays the Pervasive components on your system with version information, usage status, size, and location.
- Identifies duplicate components on your system.
- Tests your network communications to verify connectivity.
- Tests the transactional engine to verify connectivity to the MKDE.
- Tests the relational engine to verify connectivity to the SRDE.

Compared with previous versions of Pervasive.SQL, Pervasive System Analyzer replaces and extends the former SmartScout and InstallScout utilities.

For more information about Pervasive System Analyzer features, see the following sections:

- "Why Use PSA?" on page 7-3
- "PSA Tests in Detail" on page 7-3
- "Frequently Asked Questions" on page 7-7.
- "Introductory Network Information" on page 7-11
- "PSA GUI Visual Reference" on page 7-12
- "PSA Tasks" on page 7-15

Why Use PSA?

Here are some scenarios where you might use PSA after installation:

- You installed another application based on Pervasive.SQL and now one or more of your Pervasive-based applications are no longer functioning.
- You are encountering network errors and wish to test your client's connectivity to a machine running a Pervasive.SQL server.
- Your Pervasive-based application is not functioning correctly and you wish to test the functionality of the Transactional or Relational interface components to the Pervasive.SQL engine.
- You wish to restore Pervasive components archived previously by PSA.
- If you are an existing customer, you may want to archive components currently in use on your system in preparation for another install. PSA runs automatically during the installation of Pervasive.SQL 2000 SP3 and higher.

PSA Tests in Detail

This section contains more information about each functional area of PSA:

Analyze and Archive

This test is performed automatically during install, but can also be run later. You need to have administrative rights on the system in order to archive components.

Before copying new files, PSA scans your system for previous versions of Pervasive products and allows you to archive them so that they do not interfere with your new installation.

The archive and restore features are not meant as a replacement for the Add/Remove Programs dialog in Windows, which is the normal way to remove software. Archiving provides a way in which you can save a product for later restoration if you encountered issues. For example, if you were installing a newer Pervasive.SQL service pack and wanted a way in which to restore the previous service pack if you encountered issues.

If there is another archive of the same product, PSA will ask you if you want to overwrite the archive. If you choose to overwrite, the newer archive will replace the older one. PSA does not differentiate between service pack levels. Only one version of each major release

can be archived at one time. For example, you can archive both Pervasive.SQL 7 and Pervasive.SQL 2000i SP3, but you could not archive both Pervasive.SQL 2000i SP3 and Pervasive.SQL 2000i SP4.

If you archive an entire product, then the configuration settings in the Windows registry are saved as well and can be restored with the whole product. Configuration settings are not saved if you archive a portion of a product.

Note When configuration settings are archived from the Windows registry, they are copied and not deleted.

Restore Archives

If the Pervasive System Analyzer (PSA) found Pervasive archives on your system, you can select and restore these archives using this option. You need to have administrative rights on the system in order to restore archives.

All archived files are written to the `pvswarch` subdirectory of the drive that was selected when the archive was performed.

For example, if the archive was performed to drive D:\ then Pervasive archived files are stored at:

```
D:\PVSWARCH\
```

PSA scans this directory and displays the components available for restoration as shown in the following screen.

If you do not wish to restore all components, then select Restore selected files and choose the components that you wish to restore to their previous location.

If there is a version of Pervasive.SQL installed at the restore destination directory, you should uninstall or archive the active Pervasive.SQL product first before restoring the archived component.

Delete Components or Archives

This module allows you to delete files previously archived by PSA, or to delete components and registry settings made by Pervasive products. You need to have administrative rights on the system in order to delete components and archives.

View Components

This module allows you to view all Pervasive components and any other files you specify. You can add additional components to the search list. Their versions, usage status, file size, and location is displayed. If duplicate files are found, they are highlighted so you can resolve the potential conflict.

The View Components module allows you to scan a machine for Pervasive components and see which of these components are loaded.

Testing Active Installations

This module allows you to test the network connectivity between a Pervasive.SQL client and server, the functionality of the transactional engine, and the functionality of the relational engine. In the case of errors, PSA gives you detailed troubleshooting information to help you resolve the problem. The following sections discuss the three areas of testing.

Network

The Network Communication tests are performed in order to verify that your client or workstation can communicate with network protocols to reach the machine on which your Pervasive.SQL server is installed.

To run this test, specify a machine name or IP address where a Pervasive.SQL engine is running.

Using the **Advanced Settings** screen, you can select the protocols you would like to test as well as the number of stress test messages to send. By default, the network test connects using any available protocol that is installed on the system and configured for use in Pervasive.SQL by way of the "Supported Protocols" configuration parameter found on page 4-8 in *Advanced Operations Guide*. The default number of stress test messages is 75. See "To specify advanced settings for network test" on page 7-25 for more information.

The progress bar will complete for all selected tests. If you are running this test as part of installation and the test completely fails, then the install is aborted.

The steps involved in testing network communication are:

1 Verify Available Protocols

2 Verify Network Client Availability

3 Qualify Target Name

4 Resolve Target Location to network address

5 Verify Server Address

6 Verify Server Connection

7 Run Stress Test

Transactional Engine

This test verifies the ability of your client interface to connect to the Pervasive.SQL transactional engine (Btrieve).

When you run this test, PSA attempts to perform basic database operations that are common to most Btrieve applications. If your machine passes this test, then the following is verified:

- Your Pervasive.SQL transactional engine is running
- Your client interface components are installed correctly
- The network communication between the client and the server is functioning correctly
- Btrieve applications running on your computer should function correctly

Relational Engine

This test runs a series of relational interface operations to simulate your requester performing actual SQL database work. A dialog box displays during the tests to show progress.

This test verifies the ability of your client interface to connect to the Pervasive.SQL relational engine (SRDE).

When you run this test, PSA attempts to perform common SQL database operations. If your machine passes this test, then the following is verified:

- Your Pervasive.SQL relational engine is running
- Your client interface components are installed correctly
- The network communication between the client and the server is functioning correctly
- SQL applications running on your computer should function correctly

Frequently Asked Questions

Listed below are some frequently asked questions about using PSA.

- Q. Where is the PSA readme located? (page 7-7)
- Q. What is the default log file name? (page 7-7)
- Q. Can I use a different log file name? (page 7-8)
- Q. What is the local default location for the log file? (page 7-8)
- Q. What is the remote default location for the log file? (page 7-8)
- Q. What kind of information is contained in the log file? (page 7-8)
- Q. What happens to the information in the log file each time the utility is run? (page 7-8)
- Q. Does PSA detect itself? (page 7-8)
- Q. How do I run PSA? (page 7-9)
- Q. When would I want to use PSA? (page 7-9)
- Q. Does PSA archive itself? (page 7-10)
- Q. Does PSA detect, archive or delete my Pervasive Tango? (page 7-10)
- Q. What does "components" mean? (page 7-10)
- Q. What is an archive? (page 7-10)
- Q. What is a backup directory? (page 7-10)
- Q. Is a backup directory the same as an archive directory? (page 7-10)
- Q. What is the difference between a local machine and a remote NetWare volume? (page 7-11)
- Q. If I select to remove Pervasive components from my remote NetWare volume, what happens to the files on my local machine? (page 7-11)

Where is the PSA readme located?

The PSA readme file is named `readpsa.txt` and it is located in `<OSdrive>\Program Files\Common Files\Pervasive Software Shared\PSA` where `<OSdrive>` is the drive letter where your operating system is installed.

What is the default log file name?

The default file name for the log file is `PSALog.txt`.

Can I use a different log file name?

You can rename the file, but because information is appended to the file each time PSA runs, we recommend that you consistently use the same file name.

What is the local default location for the log file?

The default local location for the log file is `<OSdrive>\Program Files\Common Files\Pervasive Software Shared\PSA` where `<OSdrive>` is the drive letter where your operating system is installed.

What is the remote default location for the log file?

The default location for the log file is always the directory where psawizrd.exe is located on a Windows machine. Regardless of whether psawizrd.exe is used to delete or analyze a remote NetWare volume, psawizrd.exe is still local on the Windows machine.

What kind of information is contained in the log file?

The log file records any process performed by PSA and includes the associated timestamp. For example, if the Delete components or archives option is executed, the log file will list all files, registry settings, folders, icons and archives that PSA deletes.

What happens to the information in the log file each time the utility is run?

New information is appended to the log file each time PSA runs.

Does PSA detect itself?

No. Pervasive.SQL and PSA installations always install PSA to `<OSdrive>\Program Files\Common Files\Pervasive Software Shared\PSA`. PSA will not detect the copy in `<OSdrive>\Program Files\Common Files\Pervasive Software Shared`.

Note We strongly discourage having more than one copy of PSA on your system. However, if you find yourself in a situation where there are multiple copies of PSA on your machine, PSA will detect any other copies of PSA that are not in `<OSdrive>\Program Files\Common Files\Pervasive Software Shared\PSA` directory and you can choose to archive them.

How do I run PSA?

There are three methods for running PSA.

- During Installation:

  PSA automatically runs at install time to prepare your system for Pervasive.SQL product installation.

- From the Start Menu:

  Click **Start ▸ Programs ▸ Pervasive System Analyzer**

- From the Command Prompt:

  Click **Start ▸ Run**

  Type `<OSdrive>\Program Files\Common Files\Pervasive Software Shared\PSA\psawizrd` and press Enter.

When would I want to use PSA?

The following lists the most common situations where you would want to use PSA:

- You installed another application based on Pervasive.SQL and now one or more of your Pervasive-based applications are no longer functioning.
- You are encountering network errors and wish to test your client's connectivity to a machine running a Pervasive.SQL server.
- Your Pervasive-based application is not functioning correctly and you wish to test the functionality of the Transactional or Relational interface components to the Pervasive.SQL engine.
- You wish to restore Pervasive components archived previously by PSA.

- You wish to archive components currently in use on your system in preparation for another install. PSA runs automatically during the installation of Pervasive.SQL 2000 SP3 and higher.
- You wish to permanently delete Pervasive components or archives currently on your machine.
- You wish to view the Pervasive.SQL components that are loaded in memory.

Does PSA archive itself?

No, because PSA resides in a subdirectory of the Program Files folder and PSA searches for files in the pvsw\bin folder, it does not detect or archive itself.

See also "Does PSA detect itself?" on page 7-8

Does PSA detect, archive or delete my Pervasive Tango?

No, at this time PSA only detects Pervasive.SQL, Btrieve, and Scalable SQL product installations.

What does "components" mean?

Components refers to Pervasive.SQL files, icons, folders, program groups, engine configuration settings, registry settings, and licenses.

What is an archive?

An archive is any set of Pervasive files that PSA archived.

What is a backup directory?

This is the backup directory created by Pervasive installation. The backup directory resides in pvsw\bin\pvsw7.bck. These backup directories created by installation are different from archives created by PSA.

Is a backup directory the same as an archive directory?

No. An archive is any set of Pervasive files that PSA archived. A .bck backup directory is a directory with .bck extension that Pervasive installation programs creates prior to Pervasive.SQL 2000i SP3.

What is the difference between a local machine and a remote NetWare volume?

A local machine is the machine on which you are working on. A mapped drive to remote NetWare volume is a drive mapped on your local machine to a remote NetWare volume.

If I select to remove Pervasive components from my remote NetWare volume, what happens to the files on my local machine?

Nothing. PSA does not remove components on your local machine when you select Mapped drive to NetWare volume.

Introductory Network Information

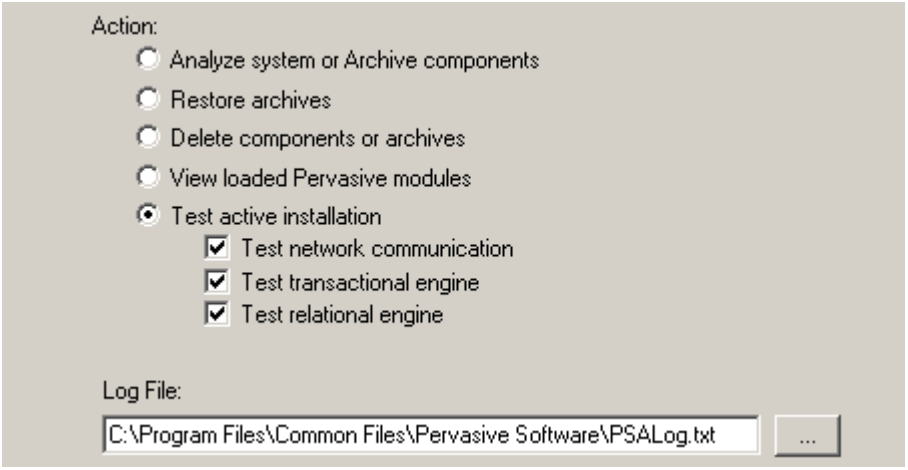
Getting Started with Pervasive.SQL (Workgroup edition) contains an appendix with basic how-to information on maintaining a successful network. If you are new to networking, you are encouraged to read this section.

Please see "Introduction to Networking" on page A-1 in *Getting Started with Pervasive.SQL* (Workgroup edition).

PSA GUI Visual Reference

The following dialog in PSA provides access to all its functionality. Click on an area of the screen to learn more about the subfunction:

Figure 7-1 PSA Main Dialog



GUI Object	Description	Related Information
Analyze system	Scans a computer for Pervasive components and allows you to archive previous versions of Pervasive.SQL software so that you can install a newer version without conflicts.	“Analyze and Archive Tasks” on page 7-16
Restore archives	Allows you to restore files previously archived by PSA into their original locations.	“Restore Archive Tasks” on page 7-17
Delete components	Allows you to permanently remove files previously archived by PSA as a group.	“Delete Components or Archives Tasks” on page 7-18
View loaded Pervasive modules	Displays all current Pervasive.SQL components and version information in a table. Also allows you to add additional components to the matrix.	“View Components Tasks” on page 7-21

GUI Object	Description	Related Information
Test active installation	This function allows you to test Pervasive.SQL installations in three ways: network connectivity, functionality of the transactional (Btrieve) database engine, and functionality of the relational (SQL) database engine. Select the checkboxes that correspond to the tests you wish to perform. **Note:** You need one of the following products to run network, transactional, and relational tests. Otherwise, these tests will be unavailable: ◆ Pervasive.SQL Client ◆ Pervasive.SQL Workgroup engine ◆ Pervasive.SQL Server engine	"Test Active Installation Tasks" on page 7-24
Test network	Test the network connectivity to a server. The network tests display detailed information about any problems encountered during the test and provides suggestions on how to remedy the problems. If you do not have Internet Explorer installed, this test is not available because the test uses an embedded HTML component to display results.	◆ "To test your network" on page 7-25 ◆ "To specify advanced settings for network test" on page 7-25 ◆ "To understand your network test results" on page 7-26
Test transactional engine	Tests the functionality of the transactional (Btrieve) database engine.	◆ "To test the transactional database engine" on page 7-27 ◆ "To understand your transactional database engine test results" on page 7-27

GUI Object	Description	Related Information
Test relational engine	Tests the functionality of the relational (SQL) database engine.	◆ "To test the relational database engine" on page 7-27 ◆ "To understand your relational database engine test results" on page 7-28
Log file specification	Allows you to specify a different log file location than the default. PSA logs detailed information on the tests it performs. You can use this log file to review tests at a later time or to forward them to Pervasive Technical Support for further review.	"To specify a different location for the PSA log file" on page 7-28

PSA Tasks

The following are the primary tasks for PSA:

Analyze and Archive Tasks

- To choose the search method for PSA (page 7-16)
- To archive components found by PSA (page 7-16)

Restore Archive Tasks

- To select the drive that contains archives (page 7-17)
- To restore archived files (page 7-17)

Delete Components or Archives Tasks

- To specify the location at which you wish to delete components or archives (page 7-18)
- To specify what to delete on a Windows machine (page 7-19)
- To specify what to delete on a NetWare machine (page 7-20)

View Components Tasks

- To select options for the View Components module (page 7-21)
- To add a component to the search list (page 7-21)
- To add a location to the search list (page 7-22)
- To understand the results of your components view (page 7-23)

Test Active Installation Tasks

- To test your network (page 7-25)
- To specify advanced settings for network test (page 7-25)
- To understand your network test results (page 7-26)
- To test the transactional database engine (page 7-27)
- To understand your transactional database engine test results (page 7-27)
- To test the relational database engine (page 7-27)
- To understand your relational database engine test results (page 7-28)

Other Tasks

- To specify a different location for the PSA log file (page 7-28)
- To view the log file at the completion of Pervasive System Analyzer tests (page 7-29)
- To determine what version of PSA you are running (page 7-29)

Analyze and Archive Tasks

➢ To choose the search method for PSA

1. Select **Analyze System or Archive Components** as shown in Figure 7-1 on page 7-12.
2. Select the search method you want the Pervasive System Analyzer (PSA) to use.
   - Quick Search - This method searches the path names included in your user and system's PATH environment variable.
   - Advanced Search - This method searches locations you specify in the Look in drop-down list box. Use the ... button to accurately select the directory location you want PSA to search. Select the Include subfolders check box to search each subfolder in the chosen directory.
3. Click **Next** to start the analysis.
4. The analysis may take several minutes. When it is complete, see "To archive components found by PSA" on page 7-16

➢ To archive components found by PSA

1. PSA displays the results of your search method ("To choose the search method for PSA" on page 7-16). If Pervasive components were found on your system, they are displayed on the results screen and are grouped by major release.
2. If you choose to archive components, the archives can be restored at a later time, if needed, using the Restore Archives option in PSA.

   If you archive all files under a major release (for example Pervasive.SQL 2000), then all the existing Pervasive components for that release and its configuration settings are archived.

   If you select a partial set of files from a release, then those files are archived but the configuration settings for that release are not copied with that archive.

Note During an install of a Pervasive.SQL engine, previous versions of the product are automatically archived for you so that your install can complete without component conflicts.

3 After selecting what files to archive, select a local drive where you would like to store the archive and click **Next** to start the archival process.

Restore Archive Tasks

➤ To select the drive that contains archives

1 Using the drop-down list, select the archive drive that contains the archives you would like to restore.

2 Click **Next** to display the archives list.

3 See "To restore archived files" on page 7-17.

➤ To restore archived files

1 PSA displays the archived Pervasive components found on your selected archive drive that are available for restoration. (See "To select the drive that contains archives" on page 7-17). The archives are grouped by major release on the results screen.

2 Choose which archives you want to restore by selecting the appropriate entries in the results list. Clear the selection of any archive that you do not want to be restored. To understand the restrictions on restoring archives, review "Notes on Restoring Archives" on page 7-18 before making your selections.

- If you choose to restore all files in a major release, such as Pervasive.SQL 2000, then all files and the configuration settings are restored to their original location.
- If you choose to restore a partial set of files from a release, then the files are restored but not any configuration settings.

3 Click **Next** to restore your selections.

If any of the files that are to be restored exist in the target directory, PSA displays a confirmation dialog before overwriting the files.

Notes on Restoring Archives

This section documents rules with regards to restoring archives.

Restoring When a Windows Server Engine is Present

When there is an existing Pervasive.SQL server engine of any version currently on your system, you cannot restore another product to that system. In this scenario, PSA displays a message that the current server engine version must be archived or uninstalled before restoring any other version can occur.

Restoring to a Remote NetWare Server

When you restore an archive to a remote NetWare server, you overwrite the Pervasive.SQL version currently on that server. PSA warns you of this before restoring to a NetWare server and allows you to cancel the operation if necessary.

Restoring Workgroup Engines and Pervasive.SQL Clients

You can restore an archive of a Workgroup engine or a Pervasive.SQL client, even if another version currently exists on your machine. However, you cannot restore the archive of either a Workgroup engine or client to a folder where another version currently exists.

You also cannot restore a Workgroup engine or client to a machine where a current Pervasive.SQL Server engine exists.

Delete Components or Archives Tasks

➢ To specify the location at which you wish to delete components or archives

1. Select **Delete Components or Archives** as shown in Figure 7-1 on page 7-12.
2. Specify the location from where you want to delete Pervasive components.
   - **Local Machine** - the machine from which you are currently running PSA.
   - **Mapped drive to NetWare volume** - Remote NetWare server linked to a client through a mapped network drive.

Note If this selection is disabled, then your Windows machine has no mapped drives to NetWare volumes. To delete Pervasive components on a remote NetWare server, map a drive to the NetWare volume where Pervasive components are installed.

3 Click **Next** to continue.

4 If you are deleting from a Windows machine, see "To specify what to delete on a Windows machine" on page 7-19. If you are deleting from a NetWare machine, see "To specify what to delete on a NetWare machine" on page 7-20

➢ To specify what to delete on a Windows machine

1 Select the items you want to delete. You can select to delete Pervasive.SQL components and registry settings or Pervasive archives.

- Delete Pervasive.SQL components and registry settings

  Permanently deletes all Pervasive.SQL files found in PATH. Deletes <drive>:\pvsw and <drive>:\bti folders and all of their contents. Deletes product registry settings, system environment variables (namely CLASSPATH, PATH and Pervasive_Path), icons, and backup (*.bck) directories created by the Pervasive installation setup program. Note: This option does not delete Pervasive Tango or archives created by PSA.

- Delete Pervasive archives

  Permanently deletes all archives created by PSA.

2 Click **Next** to delete your selections.

Note Pervasive.SQL includes the following products:

- Pervasive.SQL SDK
- Pervasive IDS
- Pervasive.SQL DataExchange
- Pervasive.SQL Workstation
- Pervasive.SQL Workgroup

- Pervasive.SQL Server Engine
- Pervasive.SQL Client
- Pervasive.SQL Install Toolkit

➤ To specify what to delete on a NetWare machine

1 Select the items you want to delete. You can select to delete Pervasive.SQL components and registry settings or Pervasive archives.

- Delete Pervasive.SQL components and registry settings

  Permanently deletes all Btrieve and Pervasive.SQL files found in the SYSTEM and ODBC folders on the specified NetWare volume. Deletes <mapped drive>:\pvsw, <mapped drive>:\bti, and <mapped drive>:\pvsw7.bck folders and all of their contents. Deletes backup (*.bck) directories created by the Pervasive installation setup program. Note: This option does not delete Pervasive Tango or archives created by PSA.

Caution Some NetWare system modules may not load correctly if the Btrieve or Pervasive.SQL components are deleted. To correct this, you must either reinstall Btrieve or Pervasive.SQL after you have deleted these components.

- Delete Pervasive archives

  Permanently deletes all archives created by PSA.

2 Click **Next** to delete your selections.

Note Pervasive.SQL includes the following products:

- Pervasive.SQL SDK
- Pervasive IDS
- Pervasive.SQL DataExchange
- Pervasive.SQL Workstation
- Pervasive.SQL Workgroup
- Pervasive.SQL Server Engine
- Pervasive.SQL Client

- Pervasive.SQL Install Toolkit

View Components Tasks

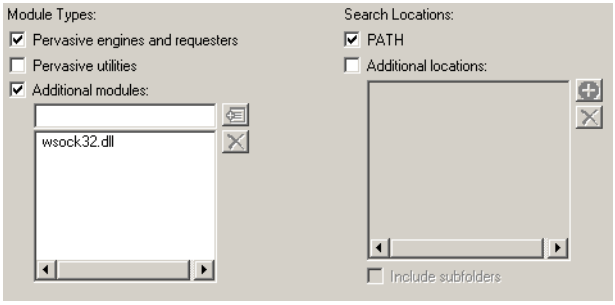
➢ To select options for the View Components module

1 Select **View Loaded Pervasive Modules** as shown in Figure 7-1 on page 7-12.

2 The selections by default resemble Figure 7-2 on page 7-21. By default, PSA displays the following components

- Pervasive engines and requesters
- Winsock DLL (networking component)

Figure 7-2 View Modules Section of PSA



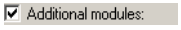
3 If you want to add components to the search list, see "To add a component to the search list" on page 7-21. If you want to add more paths to the search locations, see "To add a location to the search list" on page 7-22.

Otherwise, continue with this procedure.

4 Click **Next** to search for and display the components results list. See "To understand the results of your components view" on page 7-23.

➢ To add a component to the search list

1 Referencing Figure 7-2 on page 7-21, ensure that Additional modules checkbox is selected.



2 Enter the base component filename (without a drive letter or path) in the text box immediately below the Additional modules checkbox. For example, MYFILE.DLL

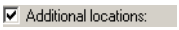
3 Click the **Add Component** button.

4 If the component you added is not in the path, add the path to the searched locations using the "To add a location to the search list" task on page 7-22.

5 Return to Step 4 of the "To select options for the View Components module" task on page 7-21

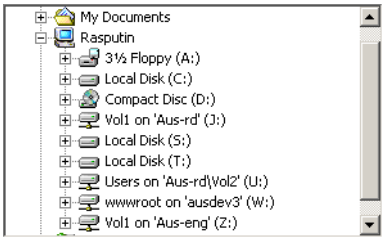
➤ To add a location to the search list

1 Referencing Figure 7-2 on page 7-21, ensure that the Additional locations checkbox is selected.



2 Click the **Add Location** button.

3 Browse to the location using the directory selector.



4 Click **OK** to add the location.

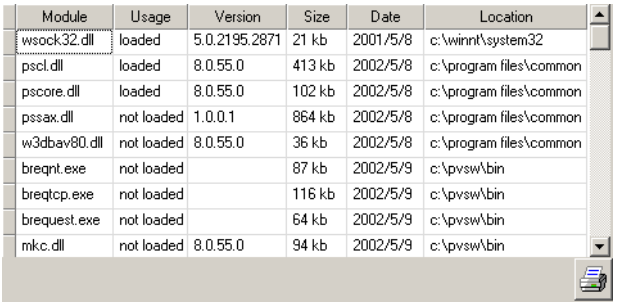
5 If you want all directories below your location to be searched as well, select the Include subfolders checkbox as shown in Figure 7-2 on page 7-21.

6 Return to Step 4 of the "To select options for the View Components module" task on page 7-21

➤ To understand the results of your components view

1 A grid resembling Figure 7-3 displays showing the components found as a result of the task "To select options for the View Components module" on page 7-21.

Figure 7-3 View Modules Results



Module	Usage	Version	Size	Date	Location
wsock32.dll	loaded	5.0.2195.2871	21 kb	2001/5/8	c:\winnt\system32
pscl.dll	loaded	8.0.55.0	413 kb	2002/5/8	c:\program files\common
pscore.dll	loaded	8.0.55.0	102 kb	2002/5/8	c:\program files\common
pssax.dll	not loaded	1.0.0.1	864 kb	2002/5/8	c:\program files\common
w3dbav80.dll	not loaded	8.0.55.0	36 kb	2002/5/8	c:\program files\common
breqnt.exe	not loaded		87 kb	2002/5/9	c:\pvsw\bin
breqtcp.exe	not loaded		116 kb	2002/5/9	c:\pvsw\bin
brequest.exe	not loaded		64 kb	2002/5/9	c:\pvsw\bin
mkc.dll	not loaded	8.0.55.0	94 kb	2002/5/9	c:\pvsw\bin

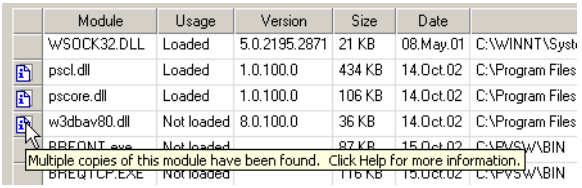
2 The grid contains the following information:

Table 7-1 Components Grid Items

Item	Description
Module	Name of the Pervasive or other component
Usage	Whether the component is currently loaded
Version	File version of the component
Size	Size of the component in bytes
Date	Date on which the component was last modified
Location	Path of the component on your system

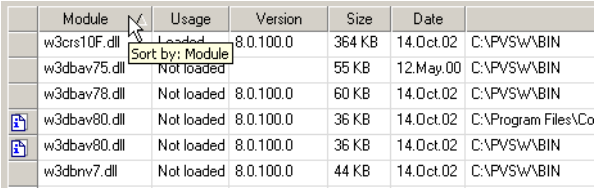
3 If files of the same name are detected, the multiple occurrences will be marked with an icon on the left side and highlighted in the list as shown in Figure 7-4. If multiple occurrences of a file are found, it does not necessarily represent a problem with your configuration. It is an informational message that you can use if you are troubleshooting an issue with components.

Figure 7-4 Multiple Occurrences of a Component Found



4 When you see multiple occurrences of a component, you may want to adjust the display so that the occurrences sort together in the list. To do this, click **Module** at the top of the display and then scroll the resulting list until the modules with the icon display appear. You should then see the multiple modules on successive rows as shown in Figure 7-5.

Figure 7-5 Sorting Multiple Occurrences of a Component



5 After you have sorted the modules by name, you can determine whether the multiple modules have the same version, and which ones are loaded in memory. If you see two Pervasive components both marked with the same version such as 8.0.xxx.0, you may want to check to see which is loaded in memory and that its location is where you expect.

Test Active Installation Tasks

Pervasive System Analyzer can test your system in these ways:

- To test your network (page 7-25)
- To specify advanced settings for network test (page 7-25)
- To understand your network test results (page 7-26)
- To test the transactional database engine (page 7-27)
- To understand your transactional database engine test results (page 7-27)

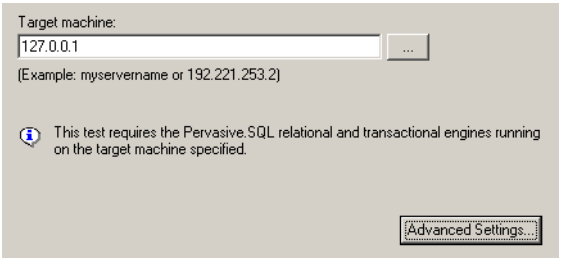
- To test the relational database engine (page 7-27)
- To understand your relational database engine test results (page 7-28)

➤ To test your network

1 Select **Test Network** as shown in Figure 7-1 on page 7-12.

2 Enter the machine name or TCP/IP address of the machine you wish to test as shown in Figure 7-6. The machine can be the machine where you are currently located or a remote machine.

Figure 7-6 Selecting the Machine on which to Run a Network Test


Target machine:
127.0.0.1
...
(Example: myservername or 192.221.253.2)
This test requires the Pervasive.SQL relational and transactional engines running on the target machine specified.
Advanced Settings...

3 If you want to control how the tests are run, click Advanced Settings and see "To specify advanced settings for network test" on page 7-25. Otherwise, continue with this procedure.

4 Click **Next** to start the test.

5 PSA runs a series of tests and displays the results. See "To understand your network test results" on page 7-26.

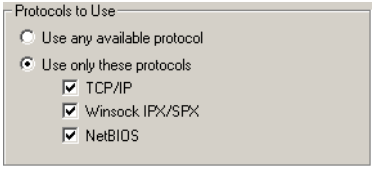
➤ To specify advanced settings for network test

1 By default, PSA uses the first available protocol. You can use this screen to force PSA to use one or more protocols. To force PSA to use protocols, select the **Use Only These Protocols** button.

2 Use the checkboxes to select the protocols you want PSA to test as shown in Figure 7-7.

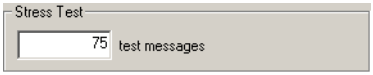
Note The NetBIOS protocol is not supported on Pervasive.SQL Server engines. The SPX protocol is not supported on Pervasive.SQL Workgroup engines.

Figure 7-7 Selecting Protocols for Network Tests



3 After successfully connecting using any protocol, PSA will send one or more stress test messages to ensure that the connectivity is sound. Specify the number of stress test messages by entering the number in the appropriate field as shown in Figure 7-8. The default number is 75, and while you can specify a maximum of 2,147,483,647 messages, a much smaller number of messages should be sufficient in most situations to ensure that your network communication is reliable.

Figure 7-8 Selecting Number of Stress Test Messages



4 Click **OK** to save the changes.

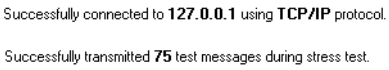
5 Return to Step 4 of the "To test your network" task on page 7-25.

➢ To understand your network test results

1 PSA runs several network tests and displays troubleshooting suggestions when problems occur.

2 If you have a successful test, the results should look like Figure 7-9.

Figure 7-9 Successful Network Test Screen


Successfully connected to 127.0.0.1 using TCP/IP protocol.
Successfully transmitted 75 test messages during stress test.

3 If your test results in issues to be resolved, then the PSA window should list the issues along with tips on how to resolve it. Read the information contained in the window, scrolling if necessary, to understand what steps need to be taken. The tips are also recorded in the PSA log file.

➤ To test the transactional database engine

1. Select whether or not you wish to run the Transactional Engine Test. Accepting the default response runs the test. This test runs a series of Btrieve operations to simulate your requester performing actual Btrieve database work.
2. Select or enter the path name for the samples directory on the database engine server. You can browse to the samples directory by clicking the button to the right of the text field.

3. Select the operations you want to perform against the transactional engine.
   - Create Data File (write access required). If the data file cannot be created, PSA attempts to load the sample database SAMPLE.BTR.
   - Read Data File
   - Update Data (write access required)
   - Insert Data (write access required)

If these steps are successful, you have a working Btrieve requester that has connectivity to the server.

➤ To understand your transactional database engine test results

A dialog box displays during the tests to show progress.

The dialog box displays the Transactional Engine test results.

Note: The progress bar completes for each transactional operation that runs. PSA displays a check mark if the operation is successful and an "x" for every operation that fails.

➤ To test the relational database engine

1. Select whether or not you wish to run the Relational Engine Test. Accepting the default response runs the tests. This verifies that the SQL Connection Manager on the remote Pervasive.SQL Server Engine is communicating properly.

2 Enter the Server Name where the DSN resides. You can browse to find the server name by clicking the button to the right of the server name field.

3 Enter the Server Data Source Name (DSN) for the data source you would like to test.

4 Select the operations you want to perform against the relational engine.

- Create Table
- Read Data
- Update Data
- Insert Data

If these steps are successful, you have a working Relational Interface requester that has connectivity to the server.

➤ To understand your relational database engine test results

A dialog box displays during the tests to show progress.

This dialog box displays the Relational Engine Test results.

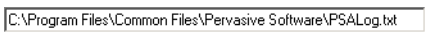
Note The progress bar completes for each relational operation that runs. PSA displays a check mark if the operation is successful and an "x" for every operation that fails.

Other Tasks

➤ To specify a different location for the PSA log file

1 On the main dialog page, enter the path to your PSA log file in the provided box.

Figure 7-10 Specifying the PSA Log File Name and Location



2 If you want to search for the directory instead, click ... to the right of the dialog box.

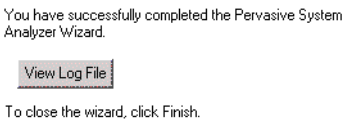
Figure 7-11 Searching for PSA Log File Path

3 Click **Next** to continue with the tests you selected.

➤ To view the log file at the completion of Pervasive System Analyzer tests

1 Click **View Log File** to display the PSA log file you specified for analysis details.

Figure 7-12 Viewing the PSA Log File



2 A summary of the tasks PSA completed is listed for you. Please restart your computer if you are instructed to do so

➤ To determine what version of PSA you are running

1 Using Windows Explorer or My Computer, browse to the location of your PSA installation.

The drive location will be the same as the location of your Windows operating system. Assuming Windows is installed on C, the PSA installation folder would be:

```
C:\Program Files\Common Files\Pervasive Software Shared\PSA
```

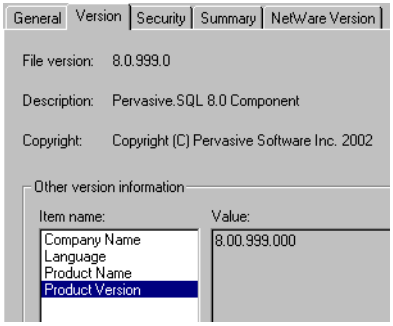
2 Find `psawizrd.exe` in the folder.

3 Right-click on the icon for `psawizrd.exe` and select **Properties**.

4 Click the **Version** tab and select the **Product Version** item.

5 Read the version on the right column as shown in Figure 7-13.

Figure 7-13 Viewing the PSA Version



chapter

8

Linux Supplementary Documentation

Additional Information on Linux Utilities

This chapter discusses information specific to Pervasive.SQL V8 for Linux. It also maps out which sections in the existing documentation that are not used with the Linux product.

The chapter contains the following sections:

- "User Manual Exclusions for Linux" on page 8-2
- "Man Pages" on page 8-4
- "Available Utilities" on page 8-5

User Manual Exclusions for Linux

Because the Linux platform is unique, Linux users are advised to look through this section when referring to the *Advanced Operations Guide* (and to the rest of this guide). This section contains exclusions and differences that apply only to the Linux release of Pervasive.SQL.

Utilities

All Win32 utilities must be run from a Windows client. If a utility must be run at the server, then its Linux equivalent will be found in this manual.

Security and user account issues relating to Linux utilities are documented in the Linux section of *Getting Started with Pervasive.SQL (Server edition)*.

Changes and Exclusions

The following sections of the *Advanced Operations Guide* do not apply to Linux. Any necessary explanations are given below.

- The sections, "Understanding the Pervasive Component Architecture" of the *Advanced Operations Guide* regarding "Overview of Smart Components," "Component Identification," or "Unique Component Naming" do not apply to Pervasive.SQL V8.
- The section, "Understanding the Pervasive Component Architecture" of the *Advanced Operations Guide* regarding "Pervasive.SQL Event Logging" is different for Pervasive.SQL V8 on Linux.

  Pervasive.SQL V8 uses the standard Linux logging system. Depending on the configuration of `/etc/syslog.conf`, messages are sent to the `syslogd` daemon, which does one of the following:

  - logs it in an appropriate system log
  - writes it to the system console
  - forwards it to a list of users
  - forwards it to `syslogd` on another host over the network

  More information can be found in the man pages for `syslogd` and `syslog.conf`.

- The section, “Configuration Reference” of the *Advanced Operations Guide* regarding the settings for “System Cache (Windows/Linux engines only)” and “Accept Remote Request (Windows/Linux engines only)” are ignored in Pervasive.SQL V8 for Linux.
- The chapter, “Manipulating Btrieve Data Files with Maintenance” of the *Advanced Operations Guide* works only on the client for Pervasive.SQL V8.

Man Pages

The following man pages are available for the Pervasive.SQL V8 for Linux product:

```
btadmin
butil
clilcadm
dbmaint
dsnadd
mkded
sqlmgr
```

To make these man pages available, add `$PVSW_ROOT/man` to your MANPATH environment variable. If you need more detailed information on a utility or application, see "Available Utilities" on page 8-5.

Note Check the man pages for the most current information. Every effort is made to ensure that the information in this guide matches that in the man pages. On occasion, last-minute changes may be included in the man pages after this guide has been published.

Available Utilities

The following utilities are available in Pervasive.SQL V8. These are specific to the Linux product and are not found in the *Advanced Operations Guide*.

- btadmin - Creates and administers database user names and passwords
- butil - Repairs and manipulates data files
- clilcadm - Applies and administers user licenses.
- dbmaint - Creates and administers named databases
- dsnadd - Creates and administers Engine DSNs on the server
- mkded - MicroKernel Database Engine daemon
- sqlmgr - SQL Relational Database Engine daemon

btadmin

Description

The btadmin utility is used to create and update the flat file btpasswd, which stores user names and passwords for authentication of Pervasive.SQL users. Users given administrator rights can monitor engine status and configure the engine remotely.

Synopsis

`btadmin [ -p` *password*`] [a+] [a-] [-r]` *username*

Options

`-p`	Specify the password. If this option is not specified, you will be prompted to enter the password.
`a+`	Gives administrator rights for this user.
`a-`	Removes administrator rights for this user.
`-r`	Remove user name from btpasswd file.

username
Creates or updates the username in the `btpasswd` file. If *username* does not exist in this file, an entry is added. If it does exist, the password is changed.

See Also

`mkded`(1), `butil`(1)

Notes

To administer the engine from a remote workstation, you must supply a user name and password. Upon initial installation of Pervasive.SQL V8, the supplied default is `admin` with an empty password.

Use `btadmin` to add more administrators:

```
% btadmin [-p password] [a+] username
```

This utility creates a record in `btpasswd` for user *username* with password *password* (if option `-p` is not used, then you will be asked

to enter a password). If a user already exists, then his password is changed as specified.

By default a user is created without administration permissions. You can use the `a+` option to give administration rights to the user. You can remove this right by using `a-`.

To remove a user record from the password file, enter:

```
% btadmin -r username
```

Every time the `btpasswd` file is changed, the previous version is backed up to `btpasswd-`.

butil

Description

The Pervasive.SQL Maintenance Utility, or `butil`, is a command line utility that performs command file and data manipulations on a MKDE file.

The maintenance utility performs the following file and data manipulations:

- Starts and stops continuous operations for use in performing server backups.
- Recovers changes made to a file between the time of the last backup and a system failure.
- Imports and exports ASCII, unformatted, and SDF sequential data.
- Copies data between data files.
- Returns MKDE version information.

Continuous operation is an MKDE feature that enables you to back up files while they are in use by Pervasive.SQL-based applications. Two maintenance commands, `startbu` and `endbu`, begin and end continuous operation on a file or set of files.

Synopsis

```
butil
    -clone outputFile sourceFile [/Oowner]
    -clrowner sourceFile [/Oowner]
    @commandFile [commandOutputFile]
    -copy sourceFile outputFile [/Oowner1] [/Oowner2]
    -create sourceFile descriptionFile [Y | N]
    -drop sourceFile <keyNumber | SYSKEY> [/Oowner]
    -endbu </A | sourceFile | @listFile>
    -index sourceFile indexFile descriptionFile [/Oowner]
    -load unformattedFile outputFile [/Oowner]
    -recover sourceFile unformattedFile [/Oowner]
    -rollfwd sourceFile unformattedFile [/Oowner]
    -save sourceFile unformattedFile
    [Y indexFile | N <keyNumber | -1>][/Oowner]
```

```
-setowner sourceFile owner level
-sindex sourceFile <descriptionFile | SYSKEY>
[keyNumber][/Oowner]
-startbu <sourceFile | @listFile>
-stat sourceFile [/Oowner]
-ver
```

Options

Note Maintenance Utility command options are not case sensitive unless the option is a filename.

If you run `butil` without specifying a command option or with an invalid command option, a usage message is printed. The usage message indicates that there is an optional `/S` command line argument to `butil`. This argument, which is used to suppress screen-at-a-time printing on NetWare, is ignored under Linux.

Commands

`-clone` *outputFile sourceFile* `[/O`*owner*`]`

The clone command creates a new, empty Pervasive.SQL formatted file with the same file specifications as an existing file (including any supplemental indexes, but excluding the owner name). The new data file includes all of the defined key characteristics (such as key position, key length, or duplicate key values) contained in the existing file. Unless specified, the new data file will not have an owner name.

`-clrowner` *sourceFile* `[/O`*owner*`]`

The `clrowner` command clears the owner name of a Pervasive.SQL data file.

`@`*commandFile* `[`*commandOutputFile*`]`

The Maintenance Utility allows you to specify the `butil` command options in a *commandFile* that can be specified to `butil`. An `<end>` keyword must be placed after each command option in the file.

The following is an example command file:

```
create xface.btr xface.dsc
<end>
```

```
stat xface.btr
<end>
```

`-copy` *sourceFile outputFile* `[/O`*owner1*`] [/O`*owner2*`]`

The `copy` command copies the contents of one Pervasive.SQL formatted file to another. `Copy` retrieves each record in the source data file and inserts it into the output data file. The record size must be the same in both files. After copying the records, `copy` displays the total number of records inserted into the output data file.

`Copy` performs the same function as `recover` and `load` in a single step.

`-create` *sourceFile descriptionFile* `[Y | N]`

The `create` command generates an empty Pervasive.SQL formatted data file using the characteristics specified in the description file. If the path name is the name of an existing Pervasive.SQL formatted file, this command creates a new, empty Pervasive.SQL formatted file in place of the existing Pervasive.SQL formatted file. Any data that was stored in the existing file is lost and cannot be recovered.

`Y|N` indicates whether to replace an existing file. If you specify N but a Pervasive.SQL formatted file with the same name exists, the utility returns an error message and does not create a new file. The default is Y.

`-drop` *sourceFile* `<`*keyNumber* `| SYSKEY> [/O`*owner*`]`

The drop command removes an index from a Pervasive.SQL formatted data file and adjusts the key numbers of any remaining indexes, subtracting 1 from each subsequent key number. If you do not want to renumber the keys, you can add 128 to the key number you specify to be dropped.

`-endbu </A |` *sourceFile* `| @`*listFile*`>`

The `endbu` command ends continuous operation on a data file or set of data files previously defined for backup. Execute this command after you have issued the `startbu` command and your backup utility has finished running. To back up data files using continuous operation, first issue the `butil –startbu` command, followed by the data file or set of data files. Next, run your backup program. Then, stop continuous operation by using the `butil – endbu` command.

`/A` stops continuous operation of all the files defined for the backup.

`-index` *sourceFile indexFile descriptionFile* `[/O`*owner*`]`

The `index` command builds an external index file for an existing Pervasive.SQL formatted file, based on a field not previously specified as a key in the existing file. Before you can use the `index` command, you must create a description file to specify the new key characteristics. The records in the new file consist of the following:

- The 4-byte address of each record in the existing data file.
- The new key value on which you want to sort.

If the key length you specify in the description file is 10 bytes, the record length of the external index file would be 14 bytes (10 plus the 4-byte address).

`-load` *unformattedFile outputFile* `[/O`*owner*`]`

The `load` command inserts records from an unformatted input sequential file into a Pervasive.SQL formatted file. It performs no conversion on the data in the input sequential file. After the utility transfers the records to the data file, it displays the total number of records loaded.

Before running the load command, you must create the input sequential file and the data file. You can create the input sequential file using a standard text editor or an application; the input sequential file must have the required file format (as explained below). You can create the data file using either `butil -create` or `butil -clone`.

Records in the input sequential file must be in the following format:

- The first field must be a left-adjusted integer (in ASCII) that provides the length of the record. This field does not include the end-of-line markers. For files with fixed length records, the length you specify should equal the record length of the data file. For files with variable length records, the length you specify must be at least as long as the minimum fixed length of the data file.
- A separator (either a comma or a blank) must follow the length field.
- The record data follows the separator. The length of the data must be the exact number of bytes specified by the length field.
- An end-of-line marker must terminate each line. The end-of-line marker is not included in the length value at the beginning of the line.

`-recover` *sourceFile unformattedFile* `[/O`*owner*`]`

The `recover` command extracts data from a Pervasive.SQL formatted file and places it in a sequential file that has the same format as the input sequential file used by the `load` command. This command is often useful for extracting some or all of the data from a damaged Pervasive.SQL formatted file. The `recover` command may be able to retrieve many, if not all, of the file's records. You can then use the `load` command to insert the recovered records into a new, undamaged Pervasive.SQL formatted file.

`-rollfwd` *sourceFile* `[ /L`*dumpfile* `/W`*dumpfile* `/T`*datalength* `/E`*keylength* `/H /V /A /O`*owner*`]`

The `rollfwd` command recovers changes made to a data file between the time of the last backup and a system failure. The MKDE stores the changes in a log.

If a system failure occurs, you can restore the backup copy of your data file and then use the `rollfwd` command, which applies all changes stored in the log to your backup copy.

`-save` *sourceFile unformattedFile* `[Y` *indexFile* `| N <`*keyNumber* `| -1>] [/O`*owner*`]`

The `save` command retrieves records from a Pervasive.SQL formatted data file using the specified index path and places them in a sequential file that is compatible with the required format for the load command. `Save` generates a single record in the output sequential file for each record in the input data file. Upon completion, save displays the total number of records saved.

The Maintenance Utility performs no conversion of the data in the records. Therefore, if you use a text editor to modify an output file containing binary data, be aware that some text editors may change the binary data, causing the results to be unpredictable.

indexFile is used if you do not want to save records by using the default of the lowest key number.

keyNumber is used if you do not want to save records using the default of the lowest key number.

-1 is the specification for saving the records in physical order using the Pervasive.SQL Step operations.

`-setowner` *sourceFile owner level*

The `setowner` command creates an owner for a Pervasive.SQL formatted file and assigns an access restriction level to that owner.

For more information about owner names, see *Advanced Operations Guide*.

`-sindex` *sourceFile* <*descriptionFile* | `SYSKEY` > [*keyNumber*] [`/O`*owner*]

The `sindex` command creates an additional index for an existing Pervasive.SQL formatted file. The key number of the new index is one higher than the previous highest key number of the data file. An exception is if a `drop` command previously removed an index without renumbering the remaining keys, thus producing an unused key number. In this case, the new index receives the first unused number.

Before you can use the `sindex` command, you must create a description file to key specifications for the index.

keyNumber is used when you do not want to save records using the default of the lowest key number.

`-startbu` <*sourceFile* | `@`*listFile*>

The `startbu` command places a file or set of files into continuous operation for backup purposes. To back up files using continuous operation, first issue the `butil -startbu` command, followed by the data file or set of data files.

Next, run your backup program. Then issue the `butil -endbu` command to stop continuous operation.

When you place a data file into continuous operation mode, the MKDE creates a temporary file with the same name as the data file, but with a `.^^^` extension. Therefore, do not create multiple data files with the same names but different extensions. For example, do not use a naming scheme such as INVOICE.HDR and INVOICE.DET for your Pervasive.SQL data files.

`-stat` *sourceFile* [`/O`*owner*]

The `stat` command reports the defined characteristics of a data file and statistics about the file's contents.

`-ver`

The `ver` command returns the version number of the MKDE loaded at the server.

Variables

descriptionFile
is the path name of a description file containing the description of

the index you wish to use. Is an ASCII text file containing information the Maintenance Utility needs to perform create and index operations. Description files are made up of one or more elements, where each element consists of a keyword followed by an equal sign (=) and a value (with no space separator). Please refer to the *Advanced Operations Guide* for details of description files.

indexFile
is the path_name of the index file in which the MKDE stores the external index.

keyNumber
is the key number (other than 0) of the index specified.

level
is the type of access restriction for the data file. The possible values for this parameter are as follows:

0 - Requires an owner name for any access mode (no data encryption).

1 - Permits read access without any owner name (no data encryption).

2 - Requires an owner name for any access mode (with data encryption).

3 - Permits read access without an owner name (with data encryption).

listFile
is the name of the text file containing the pathnames of files to be included in either the `endbu` or `startbu` operations. These pathnames must be separated by a space or end-of-line marker.

outputFile
is the path name of the data file into which you want to insert records. The output file can be empty or have existing data.

owner
is the owner of any specified files, if any. Can be used more than once in a command line when more than one file is specified. For example, `copy sourceFile outputFile [/Oowner1][/Oowner2]`. The MKDE enables you to restrict access to a file by specifying an owner name. Since owner names are optional, the files you use with this utility may or may not require an owner name. Owner names are case sensitive.

sourceFile
is the path_name of an existing data file, except when using the `create` command, which creates an empty data file. Generally refers to a Pervasive.SQL file.

unformattedFile
is the pathname of an ASCII sequential file.

Examples

- The following command creates a Pervasive.SQL formatted file named `patients.btr` using the description provided in the `BUILD.dsc` description file.

```
% butil -create patients.btr BUILD.dsc
```

- The following command copies the records in `patients.btr` to `newpats.btr`. The `patients.btr` input file does not require an owner name, but the `newpats.btr` output file uses the owner name Pam.

```
% butil -copy patients.btr newpats.btr /O /OPam
```

- The following command creates the `newapp.btr` file by cloning the `patients.btr` file.

```
% butil -clone newapp.btr patients.btr
```

- The following command clears the owner name for the `newpats.btr`. The owner name for the file `newpats.btr` is Pam.

```
% butil -clrowner patients.btr /OPam
```

- The following command sets the owner name for the `newpats.btr` to Ron with a restriction level of 1.

```
% butil -setowner patients.btr /ORon 1
```

- The following command creates an external index file called `newpats.idx` using a data file called `patients.btr`. The `patients.btr` file does not require an owner name. The description file containing the definition for the new key is called `NEWidx.dsc`

```
% butil -index patients.btr newpats.idx
NEWidx.dsc
```

The description file shown below defines a new key with one segment. The key begins at byte 30 of the record and is 10 bytes long. It enables duplicates, is modifiable, is a string type, and uses no alternate collating sequences.

```
position=30 length=10 duplicates=y modifiable=y
type=string alternate=n segment=n
```

The following two examples illustrate how to use the save command to retrieve records from a data file.

- The first example uses the `newpats.idx` external index file to retrieve records from the `patients.btr` data file and store them in an unformatted text file called `patients.sav`.

```
% butil -save patients.btr patients/.sav Y
newpats.idx
```

- The next example retrieves records from the `patients.btr` file using key number 3 and stores them in an unformatted text file called `patients.sav`.

```
% butil -save patients.btr patients/.sav N 3
```

- The following example loads sequential records from `patients.adr` file into the `patients.btr` file. The owner name of the `patients.btr` file is Sandy.

```
% butil -load patientsa.adr patients.btr /OSandy
```

- The following example adds an index to the `patients.btr` file. The name of the description file is `suppidx.dsc`. The owner name of the `patients.btr` is Ron.

```
% butil -sindex patients.btr suppidx.dsc /ORon
```

See Also

`syslogd(1)`

API Programmer's Reference – describes the Pervasive.SQL API

`$PVSW_ROOT/doc/readme.html` – contains useful configuration information and release notes (`$PVSW_ROOT` denotes the directory where Pervasive.SQL for Linux is installed - by default it is `/usr/local/psql`)

clilcadm

Description

The clilcadm utility manages the user count licenses on your server.

Note This utility can only be run by user accounts belonging to group `pvsw`. See *Getting Started with Pervasive.SQL (Server edition)* for information on Pervasive.SQL Linux utilities and user accounts.

Synopsis

`clilcadm -a` *key* `| -h | -i [`*key*`] | -r` *key*

Options

`-a` *key*	Adds a specified license key.
`-h`	Displays help information for the `clilcadm` utility.
`-i [`*key*`]`	Displays information about a specific key. If no key is specified, information about all installed keys is displayed.
`-r` *key*	Removes a specified license key.

See Also

`butil(1), btadmin(1), syslogd(1)`

Pervasive.SQL User's Guide – documents License Administration in detail.

dbmaint

Description

The dbmaint utility manages named databases.

Note This utility can only be run by user accounts belonging to group `pvsw`. See *Getting Started with Pervasive.SQL (Server edition)* for information on Pervasive.SQL Linux utilities and user accounts.

Synopsis

```
dbmaint a | d | l [-nDbname] [-a] [-b] [-i] [-e]
    [-ldictpath] [-ddatapath]
```

add new database name	a -n*Dbname* [-b] [-i] [-e] [-l*dictpath*] [-d*datapath*]
delete database name	d -n*Dbname*
list database names	`l [-a]`

Options

Commands

`add, a`	add dbname
`del, d`	delete dbname
`list, l`	list dbnames

Options

`-b`	create bound database
`-i`	create database with relational integrity
`-e`	do not create dictionary files for database
`-n`*DBName*	specify database name
`-l`*dictpath*	specify dictionary path

`-d`*datapath*	specify datapath
`-a`	show detail information about dbnames in database list

Examples

To create DBName TEST with relational integrity type:

```
% dbmaint a -i -nTEST
```

Note Unless a datapath is specified, the new database will be in the default location, $PVSW_ROOT/data. Likewise, if a dictionary path is not specified, the dictionary will be created in the default location.

To delete the same database, type:

```
% dbmaint d -nTEST
```

To list all DBNames with full information, type:

```
% dbmaint l -a
```

See Also

`dsnadd(1), butil(1), btadmin(1), syslogd(1), smb.conf(5)`

Advanced Operations Guide – for more information about database names.

dsnadd

Description

`dsnadd` simplifies the setup of a new ODBC data source that uses the Pervasive ODBC Client Interface driver. It modifies the `odbc.ini` file by adding appropriate information about the new data source. For the latest information on `dsnadd`, see the man page.

Synopsis

To add a data source, execute the following command:

```
dsnadd -dsn=server_DSN -db=DB_name
    [-ini=inifile] [-desc=DSN_description]
    [-drv-desc=driver_description]
```

or

```
dsnadd -dsn=myDSN -desc=datasource
    -db db_name -host=psqlhost -sdsn=svDSN
```

myDSN
is a name you want to assign to the new data source.

datasource
is any string to describe the data source.

psqlhost
is the name of the network host where your Pervasive.SQL is installed.

svDSN
is the name of the data source on the Pervasive.SQL host.

Options

`-help`	gives help on the `dsnadd` utility
`-odbc-ini, -ini`	ODBC.ini file name (`$HOME/.odbc.ini`)
`-dsn-name, -dsn`	Data Source Name
`-dsn-desc, -desc`	DSN description `[[`*driver_description*`:]`*host*`/`*DSN*`]`
`-srv-host, -host`	server host name
`-srv-port, -port`	server port number [1583]
`-sdsn`	server DSN [*localDSN*]
`-drv-path, -drv`	`[/usr/local/psql/lib:$HOME/lib]`
`-drv-desc`	driver description [Pervasive ODBC Client interface]

Examples

To add a Client DSN named `localDSN` that references an Engine DSN named `remoteDSN` on a machine named `server12`, enter the following command:

```
dsnadd -dsn=localDSN -desc=any_text -host=server12
    -svr-dsn=remoteDSN
```

See Also

`btadmin(1), dbmaint(1)`

mkded

Description

mkded is the command used to start and stop the Pervasive.SQL transactional server, including the underlying MKDE, as either a daemon process or in console mode. (Note that the primary way to start or stop the database engines is with the shell script psql. See "Starting and Stopping the Database Engine on Linux" on page 2-9.)

In daemon mode, Pervasive.SQL runs as a background daemon process. As a daemon, mkded relinquishes the control terminal and becomes the owner of the process group. Informational, warning, and error messages produced by the MKDE are printed to the console window (/dev/console), or to the system log (see syslogd(1)) if the MKDE cannot write to /dev/console.

When Pervasive.SQL is started in console mode, a >> prompt will be displayed allowing you to enter interactive commands. Console mode is functionally equivalent to daemon mode, except that MKDE messages are printed to standard out instead of to /dev/console (or the system log).

While the Pervasive.SQL database server is running, the butil Maintenance Utility can be executed to perform a variety of common file and data manipulation operations. For example, run butil -ver to print information on the Pervasive.SQL version that is currently running.

Synopsis

```
mkded [ -start | -stop | -console | -help ]
```

Options

The options described below are supported by mkded. These options are case-insensitive.

-start	Runs the Pervasive.SQL database server as a daemon process.
-stop	Terminates execution of Pervasive.SQL engine that was previously started in daemon mode with the -start command line argument.

`-console`	Runs the Pervasive.SQL database server in console mode. When Pervasive.SQL is started in console mode, a '>>' prompt will be displayed allowing you to enter interactive commands. The following commands are supported within console mode: `status` prints client and file engine status, `quit` exits the MKDE process `?` displays help information.
`-help`	Provides an informative message on options to the mkded command.

Configuration

Pervasive.SQL for Linux uses a bti.ini file to obtain MKDE configuration parameters. On startup, the MKDE determines the location of this file as follows:

1. If the BTIINI environment variable is defined, then look for bti.ini in the directory specified by this environment variable.
2. If (1) does not apply, then look for bti.ini in the current working directory.
3. If (1) and (2) do not apply, then look for bti.ini in the directory where the Pervasive.SQL database engine executable (`mkded`) resides.
4. If (1), (2), and (3) do not apply, then look for bti.ini in all directories contained in the $PATH environment variable.
5. If (1), (2), (3), and (4) do not apply, then create a default bti.ini in the current working directory. When the MKDE creates a default bti.ini file, an informational message is printed by the MKDE.

The configuration parameters that can be specified in bti.ini are described below. A sample configuration file can be found in `$PVSW_ROOT/samples/config/bti.ini`, where `$PVSW_ROOT` indicates the directory where Pervasive.SQL for Linux is installed.

The `bti.ini` file may contain several sections, denoted by section headings, each containing a set of configuration parameters. For this release, only the [MicroKernel] section is supported (although a [Database Names] section is created by default in the bti.ini file).

The MicroKernel performs a case-insensitive search for the name and value of each configuration parameter entry in the bti.ini file. Some entries can have more than one value. In such case, they are separated by a vertical bar (|). Values that are generic (such as, pathnames) are enclosed in angular brackets (<>).

See Also

`butil(1), btadmin(1), syslogd(1), smb.conf(5)`

API Programmer's Reference – describes the Pervasive.SQL API

$PVSW_ROOT/doc/readme.html – contains useful configuration information and release notes ($PVSW_ROOT denotes the directory where Pervasive.SQL V8 is installed).

Notes

Daemon Mode

To start the Pervasive.SQL database server as a daemon process (assuming $PVSW_ROOT/bin is in your path) run

```
% mkded -start
```

As a daemon, `mkded` relinquishes the control terminal and becomes the owner of the process group. To stop the database server process, run

```
% mkded -stop
```

Note The package installation script automatically launches `mkded` and incorporates its execution into rc scripts for automatically starting and stopping the database server as part of the system boot sequence.

Most MKDE errors are handled by returning status codes to applications that call the BTRV() and BTRVID() functions. Occasionally, special informational, warning, and error messages are printed directly by the MKDE (for example, creation of default configuration file, roll forward after for crash recovery, unable to open directory for log files). When running in daemon mode, the MKDE prints these messages to the Linux system's console window (/dev/console). You can bring up a console window on your system using

```
% xterm -C &
```

Depending on your system configuration, root access may be required to bring up a console window. If the MKDE is unable to print to /dev/console, then it will print messages to the Linux system log file. You can find out more about the system log by running `man syslogd`.

Console Mode

To start the Pervasive.SQL database server in console mode (assuming $PVSW_ROOT/bin is in your path) run

```
% mkded -console
```

After executing this command, a >> prompt will be displayed allowing you to enter interactive commands. The `status` command prints client and engine status. Entering `quit` will exit the MKDE process.

Console mode is functionally equal to daemon mode, except that MKDE messages are printed to standard out instead of /dev/console (or the system log).

sqlmgr

Description

`sqlmgr` is the command used to start and stop the Pervasive.SQL relational engine as either a daemon process or in console mode. To start `sqlmgr`, bti.ini should be placed into directory `/usr/local/psql/etc`. This file contains ODBC settings and description of ODBC DSNs for the server. (Note that the primary way to start or stop the database engines is with the shell script `psql`. See "Starting and Stopping the Database Engine on Linux" on page 2-9.)

Port and network protocol in `bti.ini` should be specified as below:

```
[SQLManager]
MgrPort=1583
MgrUseTransport=TCP
```

Each DSN should be described in `odbc.ini` as follows:

```
[DSN name]
Driver=/usr/local/psql/lib/libsrde.so
Description=Test Pervasive database
DBQ=DBName
```

To create a DBName see "dbmaint" on page 8-18.

Synopsis

```
sqlmgr [ -start | -stop | -console ]
```

Options

sqlmgr -start	starts sqlmgr
sqlmgr -stop	stops sqlmgr
sqlmgr -console	starts sqlmgr in console mode

Examples

1 SQLMGR required settings:

```
; configure sqlmgr port
    [SQLManager]
    MgrUseTransport=TCP
    MgrPort=1583
```

2 Server data source – the one to which remote calls are redirected:

```
; Test
    [dsn.srv]
    Driver=/usr/local/psql/lib/libsrde.so
    Description=Test Pervasive database
    ; DBName – named database
    DBQ=DEMODATA
```

In addition, each data source should be mentioned in the section [ODBC Data Sources] in `odbc.ini` such as the following:

```
[ODBC Data Sources]
    dsnName1=Pervasive.SQL data base
    dsnName2=Pervasive.SQL data base
```

An easy way to verify DBName and DSN configuration settings is to run the supplied odbctest program as user `psql`:

```
% /usr/local/psql/bin/odbctest DSN=DEMODATA
```

See Also

`butil(1), btadmin(1), syslogd(1), smb.conf(5), dbmaint(1)`

API Programmer's Reference – describes the Pervasive.SQL API

Notes

SqlmgrDaemon Mode

To start the SQL Manager as a daemon process (assuming $PVSW_ROOT/bin is in your path) run

```
% sqlmgr -start
```

As a daemon, `sqlmgr` relinquishes the control terminal and becomes the owner of the process group. To stop the database server process, run

```
% sqlmgr -stop
```

Note The package installation script automatically launches `sqlmgr` and incorporates its execution into rc scripts for automatically starting and stopping the database server as part of the system boot sequence.

SqlmgrConsole Mode

To start the SQL database server in console mode (assuming $PVSW_ROOT/bin is in your path) run

```
% sqlmgr -console
```

Console mode is functionally equal to daemon mode, except that system messages are printed to standard out instead of /dev/console (or the system log).

Press `^C` to exit `sqlmgr` console mode.

chapter

9

Basic Troubleshooting

How to Identify and Solve Common Problems

This chapter provides information for troubleshooting and resolving the most commonly-encountered problems.

- "General Troubleshooting" on page 9-2
- "Error Messages from PCC" on page 9-8
- "Frequently Asked Questions" on page 9-12

General Troubleshooting

This section provides some basic troubleshooting procedures to help you rule out possible causes for situations you may encounter. This section covers the following topics:

- *I get Error 1114 when trying to access my data* on page 9-2
- *I get an error saying "'ServerDSN' or 'DBQ' was not found in the connection string"* on page 9-2
- *I get a message saying my Engine components' version is different than my client components' version* on page 9-2
- *I can't get to my data on the server engine* on page 9-3
- *Error Messages from PCC* on page 9-8

I get Error 1114 when trying to access my data

or

I get an error saying "'ServerDSN' or 'DBQ' was not found in the connection string"

PCC can access remote server data sources (DSNs) using connections without client DSNs. Many desktop applications, such as Microsoft Excel and Microsoft Access, cannot do this. You must create a client DSN on your local computer to provide access to data on the server through the remote server DSN. To create a client DSN, follow the instructions in "Setting Up Client Access" on page 2-40. You must first make sure that a server DSN exists on the server you want to access.

I get a message saying my Engine components' version is different than my client components' version

When a client requester first connects to an engine, the client requester compares its internal router version with the value returned from the engine by a Btrieve Version (26) call. If the client version is older than the engine, a message dialog box is displayed on the client system with the message "Engine components' Version is different from Client's" along with a suggestion to run Pervasive System Analyzer (PSA). The same message is also logged in the client's PVSW.LOG file. This message is only a warning. The client is

not prevented from connecting to the engine in this situation. Pervasive Software does test older clients against newer engines, but guarantees compatibility between engines and clients only if the clients are the same version as, or newer than, the engines. When prompted by this message, if you choose not to run PSA and archive your old client components and install a newer client, you can expect the product to behave unpredictably until the client version is equal to or greater than the engine version.

I can't get to my data on the server engine

If you cannot get to data on the server engine, your most likely causes are:

- The server computer is down or the network has been interrupted
- You do not have operating system rights to access the server, or you are not logged into the correct network
- The client requester is not enabled
- The database server engine is not installed or not running
- The database server is not accepting remote connections
- The remote database does not have a DSN set up to advertise on the network
- The local client does not have a DSN to access the server
- The client or server network configuration is wrong

➢ To determine the actual cause of the failure

Follow the steps below to rule out certain root causes and narrow down the possible sources of failure.

1. From a Windows client, double-click **Network Neighborhood** and see if you can find the server computer that you want to connect to. If you can see the server, you can rule out that the server is down or disconnected from the network.
2. Next, try to map a drive to the file server or open a shared file on the server. If you can successfully connect to the file server and create a file on the mapped drive, then you can rule out lack of operating system rights. You can also rule out failure to login to the correct network. For example, if you have both NetWare and

Windows NT servers in your environment, it is possible to be logged into the NetWare network but not the Windows NT network, or vice versa. If you're not logged into a particular network, you can't access the servers on that network at all.

Note If you are trying to create a new database on the server, to use Monitor against the remote server engine, or to use Configuration against the remote server engine, you must have administrative rights on the server, or be a member of Pervasive_Admin. A simple drive-mapping or shared-file read will not tell you whether you have administrative rights. This means you may be able to connect to the file server, but you still may not be able to connect to the database engine with Configuration, Monitor, or Create Database Wizard.

3 The next possibility is that the client requester is disabled.

Choose **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center** to start PCC. Using PCC, double-click the icon that represents your local client computer. Double-click **Configuration** and choose **Client ▸ Access ▸ Use Remote MicroKernel Engine**. Make sure this setting is set to **On**.

You can now rule out the requester as the source of the problem.

4 Next, verify that Pervasive.SQL is installed and running on the target server.

On Windows NT, go to the server console and open the Services Control Panel and verify that "Pervasive.SQL (relational)" and "Pervasive.SQL (transactional)" have been started. If not, start these services.

On Windows 2000, go to the server console and open the Administrative Tools Control Panel and then double-click on the Services icon. Verify that "Pervasive.SQL (relational)" and "Pervasive.SQL (transactional)" have been started. If not, start these services.

On NetWare, enter the command BSTART or MGRSTART at the NetWare prompt. If Pervasive.SQL is not loaded, these commands load Btrieve and the SRDE, respectively. If Pervasive.SQL is already running, you receive the message "Modules already loaded."

On Linux, type the following command at the Linux prompt on the server where the database engine is installed:

```
ps -e | egrep 'mkded|sqlmgr'
```

If the output from the command returns at least one line containing the text "mkde" and at least one line containing the text "sqlmgr," then Pervasive.SQL is running. If you do not see these lines, then you need to be logged into the root account and start the database engine by entering
`/etc/rc.d/init.d/psql start`

You can now be certain that the server engine is installed and running.

5 The next step is to ensure that the server engine is accepting remote communication requests.

In PCC, use **Configuration** to make sure that the remote database engine is configured to accept remote requests. If you are having difficulty accessing a Windows NT server engine remotely, then you must use **Configuration** at the server itself. You must have administrative permission on the server (or membership in the Pervasive_Admin group) in order to do so. Connect to the server in PCC, double-click **Configuration** for the target server, then choose **Server ▸ Access ▸ Accept Remote Request**. Be sure the value is set to **On**.

You can now rule out the possibility the server is not accepting remote requests.

6 Note: If your application uses pure Btrieve access only, without ODBC, then skip this step.

If everything checks out so far, but you still cannot get to the data you want to access, make sure a server DSN has been set up for your target data. Using PCC, connect to the server, open the **Databases** folder for that server, and inspect the databases that are present. Make sure one of the databases represents the data you want to access. If so, then a server DSN has been created for your data.

If you do not find the data you want to access, but you know it is on the server, then most likely you need to set up a DSN for the given data. You must have administrative rights on the server (or be a member of the Pervasive_Admin group) to do so.

Right-click on the **Databases** folder for the target server, and choose **New Database**. Follow the instructions in "Setting Up ODBC Database Access" on page 2-21 to set up a DSN for existing data files.

You can now rule out the server DSN as the source of the problem.

7 Note: If your application uses pure Btrieve access only, without ODBC, then skip this step.

If you have performed all the steps above and you still cannot get to your data, the next possibility is lack of a local client DSN for the remote data.

PCC can access remote server DSNs using connections without client DSNs. Many desktop applications, such as Microsoft Excel and Microsoft Access, cannot do this. You must create a client DSN on your local computer to provide access to the remote server DSN. To create a client DSN, follow the instructions in "Setting Up Client Access" on page 2-40. You must first make sure that a server DSN exists on the server you want to access.

You can now rule out the client DSN as the source of the problem.

8 The final task to perform is to ensure that your client and server are communicating on the appropriate network protocols. By default, Pervasive.SQL ships with all network protocols enabled, so connection time may be slow as it tries all protocols, but it should eventually connect. Some application vendors disable the protocols that are not typically used by their application(s).

First, determine what protocols ought to be used on your network. If you have a Linux network or a 100% Microsoft network, then your preferred protocol is TCP/IP. If you have a NetWare network, then you need to find out from your NetWare administrator whether you should be using IPX/SPX, SPXII, or TCP/IP.

Once you know what the protocol should be, you should ensure that your server is using this protocol. You must have administrative rights on the server operating system (or be a member of Pervasive_Admin) to perform this task. Using PCC, connect to the target server. Double-click **Configuration**, and click **Server ▸ Communication Protocols ▸ Supported Protocols**.

Click **...** and ensure that the correct vendor and protocol stack is listed in the **Selected** column. Immediately above that setting, choose **ODBC Connection Manager Supported Protocol**. This setting should be set to TCP/IP unless you are configuring a NetWare server that does not have TCP/IP installed.

Ensure that your client is using the same protocol. Using PCC, double-click the icon for your local machine. Double-click **Configuration**, and choose **Client ▸ Communication Protocols ▸ Supported Protocols**. Click **...** and ensure that the correct vendor and protocol stack is in the **Selected** column.

9 If you have performed all of the above tasks with no success at accessing your data, refer to "Pervasive.SQL Resources and Contacts" on page 10-1 for more ways to get help.

Error Messages from PCC

You may receive several different messages when attempting to create or connect to databases in PCC. This section explains the likely causes for some of the most common error messages. This section explains the following messages:

- *Can't retrieve database names. You don't have access rights for the operation* on page 9-8
- *Unable to connect to the specified remote server. Verify that all of the communication components are loaded on the remote server and that there are available sessions and try again* on page 9-9
- *An error was encountered while connecting to the server* on page 9-10

Can't retrieve database names. You don't have access rights for the operation

This error may occur when you are attempting to create a new database on the server. The most likely cause is that you are logged in as an operating system user that has neither administrative rights in the server operating system, nor membership in the Pervasive_Admin group on the server. Another likely cause is that you forgot to enter a user name and password.

Solution: Be sure to enter a user name and password for the remote operating system. You must have administrative rights on the server or be a member of the Pervasive_Admin group in order to create a new database on the server. "Granting Administrative Rights for the Database Engine" on page 2-10 explains how to set up the Pervasive_Admin group.

For Windows NT/2000, be sure that you are set up as a local user on the system, not a network user. Network users have a domain name and a backslash preceding the user name, such as BOSTON\GILBERT. Be sure that the user who is a member of the Administrators group or Pervasive_Admin group is a local user.

If you have checked permissions and your user login does in fact meet one of the criteria above, then you should also check to make sure that you are logged into the correct network. For example, if you generally use the NetWare client to access servers on your network, but you are attempting to create a database on a Windows NT server,

you must make sure that you are logged into the Microsoft network, not only the Novell network on your LAN. You can verify whether you are logged into the correct network by attempting to read or write to a server that you are certain uses the target operating system.

Unable to connect to the specified remote server. Verify that all of the communication components are loaded on the remote server and that there are available sessions and try again

You may receive this error when attempting to register a new remote server in PCC. There are several reasons you may receive this error:

1. You mis-typed the server name. The database client tried to connect to a server that does not exist.

   Solution: Double-check the name of the server, and make sure you can see it in your Network Neighborhood, spelled exactly how you entered it.

   If you know the server exists but you can't see it in your Network Neighborhood, make sure that you are logged into the correct network. For example, if you generally use the NetWare client to access servers on your network, but you are attempting to connect to a database on a Windows NT server, you must make sure that you are logged into the Microsoft network, not just the Novell network on your LAN. Ask your network administrator for help.

2. The server user count has expired. If you have been using a temporary license, you will get this message for connection attempts after the license has expired.

   Solution: Run the User Count Administrator to check the status of licenses installed on the server. To start the program, choose **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Other Utilities ▸ User Count Administrator**. In the window that appears, you can see detailed status information on each license that has been applied to your server. If your license has expired, purchase a permanent license from your reseller or from Pervasive Software.

3. There are no available sessions on the server. If you have a heavy load of users on the server, or if you have configured the server with a small number of sessions, you may receive this error.

Solution: Run Monitor to check the usage of sessions available on the server. You must have administrative privileges on the server (or membership in the Pervasive_Admin group) in order to do so. To start the program, choose **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Other Utilities ▸ Monitor**. In Monitor, select **Options ▸ Connect** and connect to the server in question. Then choose **MicroKernel ▸ Communications**. In the window that appears, find **Total Remote Sessions**. If the **Peak** value and the **Maximum** value are the same, then it is likely that you have run out of sessions.

You can increase the number of sessions available by using Configuration in PCC. You must have administrative permission on the server (or membership in the Pervasive_Admin group) in order to do so. Connect to the server in PCC, double-click the Configuration icon for the target server, then choose **Server ▸ Access ▸ Number of Sessions**. Set the value to a number greater than the current setting.

4 The remote database server is not running.

Solution: Make sure that the remote database engine is running, or ask your network administrator to do so.

5 The remote database server is not accepting client requests.

Solution: Use Configuration to make sure that the remote database engine is configured to accept remote requests. You must have administrative permission on the server (or membership in the Pervasive_Admin group) in order to do so. Connect to the remote server in PCC, double-click **Configuration** for the target server, then choose **Server ▸ Access ▸ Accept Remote Request**. Be sure the value is set to **On**.

An error was encountered while connecting to the server

The most likely cause of this error is using the wrong operating system user name or password in an attempt to connect to the server.

Other possible causes include:

- The operating system may be expecting the user to change his/her password on the first logon. The Create Database Wizard does not allow this, so the connection fails. On Windows NT, this situation occurs if, in the User Manager, you have selected the **User Must Change Password at Next Logon** checkbox.
- If the user is a member of another group with lesser permissions, the lesser permissions will override the greater permissions. A user always has the most restrictive permissions of any group to which the user belongs.

Solution: Double-check the spelling of the user name and the password. Make sure the user and password have been set up on the remote server operating system.

Inspect the user's account information on the server. Make sure the operating system is not expecting the user's password to be changed at the next logon. Make sure the user is not also a member of a group that has restricted permissions.

For Windows NT/2000, be sure that the user is set up as a local user on the system, not a network user. Network users have a domain name and a backslash preceding the user name, such as BOSTON\GILBERT. Be sure that the user who is a member of the Administrators group or Pervasive_Admin group is a local user.

When trying to create a stored procedure in PCC, I get this error: ODBC Error: SQLSTATE = S1000, Native error code = -4994. The record has a key field containing a duplicate value (Btrieve Error 5)

A stored procedure already exists with the same name as the one you are trying to create.

Solution: Change the name of the stored procedure. Sometimes when you create a procedure with a SQL statement, you may not get a message from PCC confirming that the statement was successful. Then, when you run the statement again, thinking that it was not created, you receive this error code. To find out whether or not the procedure was actually created, you can get a list of the defined stored procedures by running this statement: `select * from X$proc#`

Frequently Asked Questions

This section answers some of the questions that customers ask most frequently. A list of the questions is provided below:

Installation

- Will I lose my data files if I uninstall my existing version of the product, or install a new version? (page 9-15)
- What type of client install should I do—typical, custom, or network? (page 9-15)
- How can I be sure what service pack level of client I am running? (page 9-15)
- Is Pervasive.SQL supported on a Terminal Server? (page 9-15)
- Can I install Pervasive.SQL in a Failover environment? or (page 9-15)
- Can I install Pervasive.SQL in a Clustering environment? (page 9-15)
- Can I install Pervasive.SQL in a Load Balancing environment? (page 9-16)
- Can I install Pervasive.SQL on a server running Btrieve v6.x or earlier? (page 9-16)
- I installed Pervasive.SQL on my Netware 5.x server and it still says I am running the older version. What's wrong? (page 9-16)
- How do I keep my Workgroup Engine from starting up automatically when I reboot? (page 9-16)

Security

- Your security model is confusing. When do I login using an operating system user and password, and when do I login using a database user and password? (page 9-16)
- Why do I get a "log in failed" message when I have a Pervasive_Admin group defined or I have administrator rights? (page 9-17)

Documentation

- Why does my computer have a different format of online documentation than my co-worker's computer? We both installed the same software. (page 9-17)

User Counts

- How do I apply a User Count Upgrade? (page 9-18)
- How does the Workgroup engine keep track of how many people are accessing the data? If people access the data with two engines at the same time, what happens? (page 9-18)
- Does the Workgroup engine use concurrent or per-seat licensing? (page 9-18)

- Aside from licensing, is there a limit to how many users can access a Workgroup engine simultaneously? (page 9-18)

Networking

- How do I know which protocol I am using for communication? I can see other systems in Network Neighborhood but I can't get to my data. (page 9-18)

Difficulty Accessing Data

- I upgraded from Btrieve v6.x or earlier to Pervasive.SQL V8. Now I get error messages telling me that a file is inaccessible when everybody else can get to it. What's wrong? (page 9-19)
- I have files sitting on the server that are shared and yet Pervasive.SQL cannot read them. What's wrong? (page 9-19)
- I am using SQL queries to create a definition for an old table. The resulting record size is off. Why? (page 9-20)
- I want to convert my data file version from 8 back to file format version 7, 6 or 5. How do I do this? (page 9-20)

ODBC and DDFs

- How can I tell if I can use ODBC to access my data files? (page 9-21)
- How can a hard-coded filepath in a DDF be changed? (page 9-21)
- I have DDFs from Scalable SQL 3.01. Are they compatible with Pervasive.SQL V8? (page 9-21)
- What is the best way to ensure that my data dictionaries (DDFs) are safe? (page 9-22)
- How can I tell whether I have non-standard DDFs? (page 9-22)
- Can I mix and match DDFs from different databases? (page 9-23)
- What happened to DDF Builder and DDF Sniffer? (page 9-23)
- Does Pervasive Control Center have the same functions as DDF Sniffer? Namely, can PCC read a Btrieve data file without DDFs and analyze the file to help me build DDFs? (page 9-23)
- How do I set up ODBC on a NetWare server so that I can perform relational operations? (page 9-23)
- I have two similar Btrieve files, and I created a DDF for the first one. Since they are similar, can I use the same DDF on the second Btrieve file? (page 9-25)
- I have owner names set on my Btrieve files. After I created a DSN, I cannot open the files using ODBC. What's wrong? (page 9-25)
- Is there a client side requester for the SRDE? (page 9-26)
- Is ODBC the only method of access for Pervasive.SQL? (page 9-26)

- Is there a single database file housing all the data, data definitions, stored procedures, security, table relationships, and so on as in some other products? (page 9-26)
- Does the SQL engine (SRDE) have scheduler capabilities to run stored procedures or other types of scripts designed to access and affect data? (page 9-26)

Upgrading from Btrieve 6.15

- After I upgrade the database engine on NetWare, is the SQL engine (SRDE) going to interpret Btrieve calls, or is it necessary to continue running the Btrieve.nlm? (page 9-27)
- On NetWare, will the 6.15 Btrieve NLM interface with the new MicroKernel engine, or will the 8.x version of Btrieve NLM have to be loaded to continue accessing the current Btrieve files? (page 9-27)
- My current files are in a 5.x format. Will they have to be converted to be accessed by the SQL engine (SRDE)? (page 9-27)
- I have DDF files today that were used by a product called Xtrieve. They are in a 4.x file format. Can they be used by the SQL engine or will they have to be converted? (page 9-27)
- Is there a tool that replaces Xtrieve? (page 9-27)
- I expect to continue using my old applications and files and phase in new applications to access the same files through the SQL engine (SRDE). Is this a false expectation? (page 9-28)

Upgrading and Migration

- How can I migrate a Btrieve database from NetWare to Windows NT/2000 or vice versa? (page 9-28)
- Where can I find information on migration from earlier product versions to Pervasive.SQL V8? Where can I find migration and compatibility information? (page 9-29)
- When I create a table using an existing Btrieve file, the wizard displays fewer columns than there are in the Btrieve file. What's wrong? (page 9-28)

Miscellaneous

- I dumped Btrieve records to a file and now I can't read the file. What happened? (page 9-29)
- Does Pervasive.SQL take advantage of multiple processors? (page 9-29)
- How do I run Pervasive.SQL in trace mode? (page 9-30)
- Does garbage collection occur in the data files and indexes? For example, is space from deleted records recovered or reused? (page 9-30)
- Is database shadowing available, allowing a complete up-to-date second copy of the database to exist on another drive or machine? (page 9-31)

- What is the mechanism that allows the database to be backed up online? What happens if the server goes down in the middle of a backup with many open transactions? (page 9-31)

Installation

Will I lose my data files if I uninstall my existing version of the product, or install a new version?

When you uninstall Pervasive.SQL or install a new version of Pervasive.SQL, your data files and DDFs are never affected. Even when Pervasive System Analyzer archives Pervasive.SQL files, or even if you have your data files in the same directory as Pervasive.SQL files, your data files are not affected.

What type of client install should I do—typical, custom, or network?

If you are not sure, always select typical. This option performs a standardized installation, which makes it easier to troubleshoot if problems occur.

How can I be sure what service pack level of client I am running?

If you are using Pervasive.SQL V8, start Monitor or Maintenance, choose **Help ▸ About** from the menu, and check the Build Level.

Is Pervasive.SQL supported on a Terminal Server?

Support for both the Server and Workgroup engines on Terminal Server has been available since Pervasive.SQL 2000i, SP 4. Pervasive.SQL 2000i, SP 3 provided support for the Server engine. Pervasive.SQL 2000i, SP 2 provided supports only for the client software.

Can I install Pervasive.SQL in a Failover environment? or

Can I install Pervasive.SQL in a Clustering environment?

Pervasive.SQL 2000i SP3 or later can be installed into a Windows 2000 Advanced Server Cluster or into a Novell NetWare 5.1 Cluster. Earlier versions of Pervasive.SQL are not supported in a Failover or Clustered environment. Linux clusters are not supported at this time.

Can I install Pervasive.SQL in a Load Balancing environment?

That is not supported at this time.

Can I install Pervasive.SQL on a server running Btrieve v6.x or earlier?

No, you cannot run Pervasive.SQL and Btrieve 6.x on the same computer at the same time.

I installed Pervasive.SQL on my Netware 5.x server and it still says I am running the older version. What's wrong?

With Netware 5.x, you have to shut down the server and reboot for the new version of the database engine to be loaded into memory.

How do I keep my Workgroup Engine from starting up automatically when I reboot?

You must remove it from the Startup group under Start ▸ Programs.

On Windows 98, the contents of this group are located at: c:\windows\start menu\programs\startup.

On Windows NT, the contents of this group are located at: c:\winnt\profiles*user*\start menu\programs\startup

On Windows 2000, the contents of this group are located at: c:\Documents and Settings*user*\start menu\programs\startup

On Windows NT/2000, *user* can be replaced by "All Users" or any user, as appropriate.

Security

Your security model is confusing. When do I login using an operating system user and password, and when do I login using a database user and password?

This may seem confusing at first, but in fact there is only one rule: use a database login only after you have already successfully connected to the server and are attempting to access a database directly. Up until this point, you should use an operating system login.

For example, if you run Monitor or Configuration to work with a remote server engine, you are prompted for a password. In both

cases, you must supply a user name and password for an operating system account that has administrative permissions on the remote system, or an account that is a member of Pervasive_Admin. This applies also when you are creating a new database.

Once you start to work with the data itself, then you must supply a database user and password, if prompted. If database security is turned off, then you would never need a database user name or password. In this case, you would only need an operating system user and password to perform administrative tasks, as noted in the preceding paragraph.

Why do I get a "log in failed" message when I have a Pervasive_Admin group defined or I have administrator rights?

The settings for the Pervasive.SQL services can affect whether or not you have permission to log in to the machine where the database engine is running. The settings apply whether or not you use a Pervasive_Admin group. If you change the **Log on as** setting for a Pervasive.SQL service to **This account**, you must change the user rights policy **Act as part of the operating system** for the account. Otherwise, remote log in fails.

For example, both the Monitoring utility and the Configuration utility require that you log in to the operating system on the machine where the database engine is running. You will receive a message that log in failed if the account specified for **This account** cannot act as part of the operating system.

Note that even the Administrator account requires that you set the user rights policy for **Act as part of the operating system**.

See "Services Settings and Log In Authority" on page 2-13 for a complete discussion and the steps to change the user rights policy.

Documentation

Why does my computer have a different format of online documentation than my co-worker's computer? We both installed the same software.

Starting with Pervasive.SQL 2000i Service Pack 3, the type of online documentation that is installed depends on your system. Computers that support Microsoft HTML Help, also called Windows 98 Help, automatically receive that format of online documentation during the Pervasive.SQL installation. Computers that cannot support

Microsoft HTML Help automatically receive Windows 95 Help files. The actual content within each format is identical.

User Counts

How do I apply a User Count Upgrade?

Refer to the tasks discussed in the chapter "License Administrator" on page 4-1.

How does the Workgroup engine keep track of how many people are accessing the data? If people access the data with two engines at the same time, what happens?

The Workgroup engine keeps track of users just as the Server engine does. That is, each combination of a set of requesters and an application creates a unique client ID. Licenses are tracked per computer based on the unique ID for the network card. Only one engine is ever permitted to access the files at a time. The second engine to try to open the files gets locked out, because the engines open the data files in exclusive mode (non-file sharing) so that corruption cannot occur.

Does the Workgroup engine use concurrent or per-seat licensing?

Concurrent. Refer to "User Count" on page 4-3.

Aside from licensing, is there a limit to how many users can access a Workgroup engine simultaneously?

The Workgroup engine architecture is basically the same as the Server engine. There are no theoretical limits, only practical ones such as bandwidth, system processing power, and economics. Client/server licenses are less expensive than Workgroup once you reach about 7 to 10 users.

Networking

How do I know which protocol I am using for communication? I can see other systems in Network Neighborhood but I can't get to my data.

From the **Start** menu of any Windows computer with the database client installed, choose **Programs ▸ Pervasive System Analyzer**. In the Welcome screen that appears, click **Next**. In the following screen, check the box **Test Network Communications** and make sure all the other boxes are not checked. Click **Next**. In the following screen,

cancel the selected protocols that you do not want to test. Click **Browse** to select the drive that you have mapped to the installation directory (C:\PVSW by default) on the server. You must have a mapped drive; UNC names are not supported. Click **Next** to run the network tests. The results window tells you if there are any significant problems with your networking.

Difficulty Accessing Data

I upgraded from Btrieve v6.x or earlier to Pervasive.SQL V8. Now I get error messages telling me that a file is inaccessible when everybody else can get to it. What's wrong?

Use Pervasive System Analyzer to be sure that all components from previous versions of Btrieve or Pervasive.SQL have been archived. Then, make sure your configuration settings are correct. Find the file pvsw.log and check for error messages indicating a status code 8505 or 8517. These status codes indicate that attempts were made to use a local Workgroup engine to read the data files. From the Start menu, choose **Programs ▸ Pervasive.SQL V8 ▸ Control Center**. Double-click **Pervasive.SQL Engines**. Double-click the icon representing your computer. Double-click **Configuration**. Double-click **Client**. Click **Access**. In the right-hand pane, make sure the following settings are set: Use **Local MicroKernel Engine** is "Off," and **Use Remote MicroKernel Engine** is "On."

I have files sitting on the server that are shared and yet Pervasive.SQL cannot read them. What's wrong?

How are the files shared? Pervasive does not support mapping a drive letter using the Map Root under Novell, or using Redirected mapping under Microsoft, or using the hidden Admin share (C$) under Windows NT/2000.

Make sure that users have appropriate operating system login credentials to access the file server.

Run Pervasive System Analyzer and choose the Network Communications Test to be sure that you have proper connectivity.

I am using SQL queries to create a definition for an old table. The resulting record size is off. Why?

Starting with Pervasive.SQL 2000, fields that allow null values have an additional byte defined at the start of the field. This byte is the null indicator byte. You can work around this in one of two ways:

If you are using SQL statements to create a new table definition, enter the statement SET TRUENULLCREATE=OFF. For the remainder of your current session, any tables that you create will use the old record structure without the extra byte for each nullable column.

If you do not wish to use SQL statements, you can get the field sizes to align properly by creating all columns as not nullable.

I want to convert my data file version from 8 back to file format version 7, 6 or 5. How do I do this?

If the files you wish to convert are serviced by a remote Server or Workgroup engine, you must have Administrator permissions on the remote system in order to perform these tasks. You must also have a network drive mapped to the remote data files.

Using Pervasive Control Center, double-click the icon that represents the server where the data files are located. Double-click **Configuration** for that server. Double-click **Server** then click **Compatibility**. Click on **Create File Version** and set the value to the file version to which you want to convert. From the menu, choose **Edit ▸ Apply**. Restart the database engine. These changes result in new files created to be in the version selected.

From the **Start** menu, choose **Programs ▸ Pervasive.SQL V8 ▸ Other Utilities ▸ Maintenance** to start Btrieve Maintenance Utility. Within this program, choose **Options ▸ File Information Editor**. Click **Load Information** and choose the data file that you want to convert. Click **Create** and specify the name of the new, empty data file you want to create with the older version format. Click **OK** to create the file. Close the **File Information Editor** window, but do not exit Btrieve Maintenance Utility.

From the menu, select **Data ▸ Copy**. Enter the name of the source data file and the name of the target data file (your newly created file with the older version file format). Click **Execute** to copy the records into the older version file. After the copying has finished, if you need the new data file to have the same name as it did previously, save your

original data file with a different name, then save your new file using the original file name.

ODBC and DDFs

How can I tell if I can use ODBC to access my data files?

There are several ways to find out. First, look for .DDF files where the data files are located. If you see them, then most likely you can access the database using ODBC. Because it is possible to have DDF files located in a different directory, you should also use PCC to determine whether a database has been created for the data files you want to access. Finally, you can ask your application vendor whether their application uses ODBC to access the data files.

How can a hard-coded filepath in a DDF be changed?

Using PCC, right click on the table and choose **Tasks ▸ Edit table design**. Click on the **Statistics** tab. Locate the parameter **Table Location** and change the value to the file path you wish to use. From the menu, choose **File ▸ Save**.

It may appear that the path has not changed. To confirm the change, open the X$File system table and look at the Xf$Loc field for the given user table. If you cannot see the system tables in PCC, click on the **View** menu and choose **Show system tables**.

You can also use the ALTER TABLE USING statement in SQL to change the data file used by a particular table. Refer to *SQL Engine Reference* for further information.

I have DDFs from Scalable SQL 3.01. Are they compatible with Pervasive.SQL V8?

DDFs from Scalable SQL 3.01 are not compatible with Pervasive.SQL V8. You can use the CNVDDF utility provided on the Pervasive web site to convert the older DDFs to the new format. CNVDDF is a DOS utility that enables you to convert a database dictionary from Scalable SQL 3.01 to Pervasive.SQL V8 format. To run the utility, make sure that either the Btrieve DOS Box or another DOS requester is loaded on the client workstation.

The utility is located at:

http://www.pervasive.com/support/toolbox.asp

Note Converting database dictionary files in the CNVDDF utility does not modify any non-system table Btrieve files; it only modifies FILE.DDF, FIELD.DDF, and INDEX.DDF. It is recommended that you back up the original files to another directory before making any changes to the dictionary files using CNVDDF. This will enable you to restore your original DDFs.

What is the best way to ensure that my data dictionaries (DDFs) are safe?

Always keep a backup copy of your DDFs. Anytime you make changes to the runtime DDFs, be sure to make a backup copy of the DDFs before making changes. If you are turning on database security for the first time, you should make a backup copy of the dictionaries without security, and a backup copy with security.

How can I tell whether I have non-standard DDFs?

If you can edit your DDFs with a Btrieve utility, it means that you do not have standard dictionary files. A standard dictionary file does not permit direct Btrieve access. This lock out is a safety feature that ensures only the SRDE can write to the dictionary. DDFs are very special files that must remain synchronized with each other and with the data files at all times.

Standard dictionaries do not have case sensitive table names or field names. That is, the column definitions for column Xf$Name in file.ddf and column Xe$Name in field.ddf have the Case flag set, meaning the values are case insensitive.

DDFs are Btrieve files and thus can be opened and viewed (not updated) using the Function Executor. This is one way to confirm the contents of file.ddf or field.ddf.

On some non-standard dictionaries, the DDFs file.ddf, field.ddf, and/or index.ddf do not exist. Such dictionaries do not work with our products. For example, if you see a file called x$file.ddf, instead of file.ddf, you can assume your DDFs are non-standard.

Non-standard DDFs are unlikely to work properly with Pervasive Control Center or the relational engine.

Can I mix and match DDFs from different databases?

A complete set of DDF files must be considered a unit. No DDF file from one database may be intermixed with DDFs from a different database.

What happened to DDF Builder and DDF Sniffer?

DDF Sniffer and DDF Builder were added to the Pervasive product line with the acquisition of Smithware in 1998. Neither DDF Sniffer nor DDF Builder are available anymore. They were replaced by DDF Ease, which was part of the Pervasive.SQL 7 database engine. DDF Ease has since been replaced by Pervasive Control Center in Pervasive.SQL V8.

DDF Sniffer and DDF Builder used the Btrieve API to manipulate DDFs, which was less desirable than doing it through the native relational interface of Pervasive.SQL. These products contributed to issues that were difficult to isolate and fix.

Does Pervasive Control Center have the same functions as DDF Sniffer? Namely, can PCC read a Btrieve data file without DDFs and analyze the file to help me build DDFs?

Yes, it does in general, but it lacks some of the automation features offered by DDF Sniffer, such as the ability to recommend data types to use for non-indexed fields. However, PCC avoids many of the problems that DDF Sniffer and DDF Builder encountered in trying to create and maintain DDFs via the Btrieve API rather than through the relational engine. The result is that you probably need to know a bit more about databases and data types than you would with DDF Sniffer, but you will wind up with fewer problems in the long run.

For more information, see the question and answer immediately above.

How do I set up ODBC on a NetWare server so that I can perform relational operations?

Pervasive.SQL includes an ODBC manager for NetWare. All you need to do is use PCC to create a new database on the NetWare server, then create a client DSN on the client to make the remote database available to client applications.

You must have administrative permissions on the NetWare server to perform these tasks. Here are the steps in more detail:

1. Load the relational module on the NetWare server by issuing the command `mgrstart`.
2. At a Windows computer with Pervasive.SQL client installed, start PCC by choosing **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Control Center**. Double-click **Pervasive.SQL Engines**. If you do not see the NetWare server name listed, then right-click and choose **Register New Engine**. In the window that appears, type in or browse to the NetWare server where the database engine is located. Click **OK**.
3. After the server icon is displayed in PCC, double-click on the icon that represents the server. Then double-click on the **Databases** folder. If you do not see the database you want to connect to, right-click on the **Databases** folder and choose **New Database**.
4. In the screen that appears, choose **Engine** interface and enter a NetWare user name and password with administrative permissions on the NetWare server. Click **Next**.
5. In the following screen, enter the name of the database and the directory where the data files are or will be located. This directory cannot be a mapped drive or relative to the client. It must be a full path name on the server, as if you were seated at the server console.

   If you want to make existing DDFs and data files available for remote access, check **Use Advanced Settings**. If you are creating a brand new database from scratch, do not check this box.
6. Follow the prompts provided. For detailed procedures and options, see Chapter 2 of this book and/or see *Advanced Operations Guide*.

After you have created the Engine DSN on the server, you can access the database using PCC. If you want to access the database using ODBC-based applications, then you need to use ODBC Administrator to create a local client DSN on your workstation. You can do so by choosing **Start ▸ Programs ▸ Pervasive.SQL V8 ▸ Other Utilities ▸ ODBC Administrator**. In ODBC Administrator, click the **System** tab, then click **Add**. In the window that appears, select **Pervasive ODBC Client Interface** and click **Finish**. In the following

window, type in a name for the local DSN, and enter the server name. Click **Get DSN List**. In the box labeled **Data Source Name**, choose the DSN that you created on the NetWare server. Click **OK**.

You can now access the database on the NetWare server by connecting to the local client DSN you just created.

I have two similar Btrieve files, and I created a DDF for the first one. Since they are similar, can I use the same DDF on the second Btrieve file?

The answer depends on how similar the files are. If the two files differ only in the number of records, you can use the same DDF file. If there are any differences at all in the number, order, names, or types of fields or indexes, you cannot use the same DDF. In other words, you can only use the same DDF if the record structure of the two files is identical.

I have owner names set on my Btrieve files. After I created a DSN, I cannot open the files using ODBC. What's wrong?

If Btrieve files have owner names on them, you must use database security for ODBC access. Turn on database security in PCC by following these steps: right-click on the database name, choose **Properties**, and click on the **Security** tab. In the screen that appears, enter and confirm the master user password.

Caution Do not forget the Master user password. You cannot turn off security or perform administrative tasks within the database without it. You may want to make a backup copy of your DDFs before turning security on, in case you forget the password.

Next, you must grant the Master user access to the data files that have owner names defined. You can grant the access by issuing this SQL statement for each table that has an owner name:

```
GRANT ALL ON my_table 'ownername' TO Master
```

When you enter the statement, substitute the actual name of your table and the appropriate owner name for that table, as indicated above. Remember that each data file corresponds to an ODBC table. If you don't know which table corresponds to which data file, use

PCC to find out: right-click on the table in PCC, and choose **Tasks ▸ Edit Table Design**. In the Table Designer, click on the **Statistics** tab. The **Table Location** field shows you the file that is referenced by that table definition.

If security is important, then you must create users and assign permissions for all users expected to access the database. You do this by using CREATE USER, CREATE GROUP, and GRANT statements in SQL. You can also use the **Users and Groups** feature of PCC.

If security is not important to you, you can avoid creating many users and assigning privileges by granting access to PUBLIC, which means anyone on your network can access the data. You can use this statement:

```
GRANT ALL ON my_table 'ownername' TO PUBLIC
```

Is there a client side requester for the SRDE?

There is no DOS requester support for SQL applications, but the Pervasive.SQL client software for Windows includes ODBC client components allowing you to connect to a remote SRDE server engine.

Is ODBC the only method of access for Pervasive.SQL?

Definitely not! In addition to ODBC and the time-tested Btrieve API, you can also develop applications using our OLE DB provider, our JDBC driver, our pure Java interface, or our ActiveX controls.

Is there a single database file housing all the data, data definitions, stored procedures, security, table relationships, and so on as in some other products?

No. Pervasive.SQL stores data in separate files, one file per relational table definition. The meta data, such as data definitions, user/group definitions, and so on, are stored in a set of DDF files, where each file ends in the extension ".ddf."

Does the SQL engine (SRDE) have scheduler capabilities to run stored procedures or other types of scripts designed to access and affect data?

The SRDE does not have a scheduler.

Upgrading from Btrieve 6.15

After I upgrade the database engine on NetWare, is the SQL engine (SRDE) going to interpret Btrieve calls, or is it necessary to continue running the Btrieve.nlm?

Pervasive.SQL includes a new version of Btrieve.nlm. It also includes a new module named nwmkde.nlm. In the new architecture, Btrieve.nlm is a stub that calls nwmkde.nlm to perform operations. You must have both of these modules running.

On NetWare, will the 6.15 Btrieve NLM interface with the new MicroKernel engine, or will the 8.x version of Btrieve NLM have to be loaded to continue accessing the current Btrieve files?

You cannot use the 6.15 version of Btrieve.NLM with Pervasive.SQL. You must load all NLMs from the same product version.

My current files are in a 5.x format. Will they have to be converted to be accessed by the SQL engine (SRDE)?

The SRDE accesses 5.x format files through the MicroKernel. No file conversion is needed. You must convert the files to 8.x format to take advantage of new features offered by the 8.x MicroKernel. However, the SRDE requires DDFs for your data files. If you do not have DDFs, in some cases you may be required to modify your file indexes before you can create valid DDFs. The steps required to create DDFs for Btrieve files are explained in *Advanced Operations Guide*, Chapter 13.

I have DDF files today that were used by a product called Xtrieve. They are in a 4.x file format. Can they be used by the SQL engine or will they have to be converted?

They must be converted to the Pervasive.SQL DDF format using the tool DDFCNV. For more information about this tool, see "I have DDFs from Scalable SQL 3.01. Are they compatible with Pervasive.SQL V8?" on page 9-21.

Is there a tool that replaces Xtrieve?

There is no direct replacement, but you should consider using Crystal Reports for Btrieve as an excellent upgrade from Xtrieve for reporting on and querying Btrieve data.

I expect to continue using my old applications and files and phase in new applications to access the same files through the SQL engine (SRDE). Is this a false expectation?

Generally speaking, no. Pervasive.SQL is designed to allow concurrent access between SRDE applications and Btrieve applications. If you are currently using Btrieve without ODBC, you may need to make some changes to the indexes on your files before you can create DDFs that provide SRDE access to the data. The steps required to create DDFs for Btrieve files are explained in *Advanced Operations Guide*, Chapter 13.

Upgrading and Migration

How can I migrate a Btrieve database from NetWare to Windows NT/2000 or vice versa?

Shut down the Btrieve engine on the source computer and copy all the database files from the source computer to the target computer.

Install Pervasive.SQL V8 Server engine on the target computer.

Create a new database using existing DDFs and data files.

Share the server resource as appropriate.

Win32 Clients: No change

DOS Clients: If moving to Windows NT, replace BREQUEST with BREQNT or BREQTCP. If moving to NetWare, replace BREQNT or BREQTCP with BREQUEST. See also the `readdos.txt` file installed with the database engine.

When I create a table using an existing Btrieve file, the wizard displays fewer columns than there are in the Btrieve file. What's wrong?

Btrieve files contain a limited amount of information about the structure of the file. The table creation wizard can figure out some field definitions using the indexes, but after the indexes are exhausted, data segments may remain that contain more than one actual field. The wizard has no way of interpreting the contents. You must use your detailed knowledge of the record structure to split out these fields and build a table definition that matches all the fields in the record.

The procedure for this task is provided in *Advanced Operations Guide*.

Where can I find information on migration from earlier product versions to Pervasive.SQL V8? Where can I find migration and compatibility information?

Getting Started with Pervasive.SQL contains an entire chapter that provides detailed instructions on how to upgrade.

If your application uses Scalable SQL or ODBC, then you should review the *Application Migration Guide* available on the web site:

http://www.pervasive.com/library/index.asp?_shownode=MIG

Miscellaneous

I dumped Btrieve records to a file and now I can't read the file. What happened?

If you use the Btrieve Maintenance Utility to save/dump the records, the resulting file contains the binary image of each record. Unless the record consists entirely of character data, it may not be readable to the human eye.

The only way that Pervasive.SQL can dump a record in ASCII readable format, is by reading the DDFs to get a description of the total contents of the record. Btrieve only has the record length, the data type of indexes and length of the indexes. Btrieve does not have information on how to interpret the entire contents of the record.

Does Pervasive.SQL take advantage of multiple processors?

Pervasive.SQL fully supports symmetric multiprocessing (SMP) and multiple processors because it is multi-threaded and thread safe. However, Pervasive.SQL does not take direct advantage of any call specific for SMP and is not multiprocessor aware.

In an SMP environment, the operating system schedules available threads on the available processors, including threads for Pervasive.SQL. Since Pervasive.SQL is multi-threaded and thread safe, SMP can yield a performance boost for up to four CPUs. No significant advantage has been shown, however, to bypass the operating system scheduling and use SMP-specific calls to become SMP aware. This is because, in most cases, slowness occurs from disk I/O and not from CPU use.

How do I run Pervasive.SQL in trace mode?

Server

You must have administrator privileges on the machine where the engine is located that you want to run in debug mode. Using PCC, double-click **Configuration** for the desired Server engine. Choose **Server ▸ Debugging ▸ Trace Operation** and set the value to **On**. Click **Edit ▸ Apply**. You do not need to restart the engine.

See also "Trace Operation" on page 4-17 in *Advanced Operations Guide*.

Note After tracing operations, you should turn off Trace Operation, making sure to click Edit ▸ Apply when finished. You will notice slower performance if you run Pervasive.SQL in trace mode.

Win32 Client and Btrieve Win16 Client

Run the PSA network connectivity tests to verify network connectivity. See "Test Active Installation Tasks" on page 7-24. Also refer to the Pervasive Knowledge Base (http://support.pervasive.com/eSupport) for information about particular issues.

In addition, client tracing is available for troubleshooting certain types of low-level problems. Generally, low-level tracing is not required, so this type of tracing is intended for use by trained support staff. Your product vendor or Pervasive Software Support will explain how to conduct low-level client tracing.

Does garbage collection occur in the data files and indexes? For example, is space from deleted records recovered or reused?

Yes, space from deleted records is re-used on subsequent inserts. Space in files is never de-allocated back to disk. If index balancing is turned on, then unused space in index pages is also re-used. See "Converting MicroKernel Data Files" on page 15-6 in *Advanced Operations Guide* for information about using the Rebuild utility.

Is database shadowing available, allowing a complete up-to-date second copy of the database to exist on another drive or machine?

Pervasive.SQL does not contain specific functionality for this, but many customers have successfully used hardware mirrored drive arrays and solutions like Vinca's (now acquired by Legato) Standby Server to provide this functionality. Pervasive.SQL 2000i SP3 and later supports Pervasive's data replication product, Pervasive.DataExchange. Pervasive.DataExchange can be used to synchronize all or part of two or more databases.

What is the mechanism that allows the database to be backed up online? What happens if the server goes down in the middle of a backup with many open transactions?

Continuous Operations allows you to put a set of data files in a special "safe mode" so that they can be safely backed up while in use. While data files are in Continuous Operations mode, they are not modified, and special delta files store the results of any database operations. After the backup is complete, the data files must be removed from Continuous Operations mode, at which time the changes stored in the delta files are rolled into the live files.

If the server goes down while files are in continuous operations mode, the next time the data file is accessed, the database engine detects the existing delta file and rolls in the changes at that time.

You can put data files into Continuous Operations mode by using the BUTIL -STARTBU command or Maintenance utility described in *Advanced Operations Guide*.

chapter

10

Pervasive.SQL Resources and Contacts

A Guide to Pervasive.SQL Customer Information Resources

Pervasive Software strives to ensure that your experience with Pervasive.SQL is successful. This chapter describes the resources and information available to you as a valued customer of Pervasive Software.

The following variety of resources can help you get answers to your questions, troubleshoot problems, and interact with the Pervasive team as well as with other customers:

- "Printed Documentation" on page 10-2
- "Developer Center" on page 10-3
- "Pervasive.SQL Knowledge Base" on page 10-4
- "FTP Site" on page 10-5
- "Online Documentation" on page 10-6
- "Webinars" on page 10-7
- "Subscription Based E-mail Services" on page 10-8
- "DevTalk" on page 10-9
- "Newsgroup" on page 10-10
- "E-Mail Contacts" on page 10-11
- "Technical Support" on page 10-12

Printed Documentation

Pervasive.SQL V8 comes with a printed copy of *Getting Started* and *Status Codes Quick Reference*. A complete suite of online documentation is installed on Windows when you choose the **Typical** installation procedure. It is available as an option in the **Custom** installation procedure. The content is accessible through the **Start** menu:
Programs ▸ Pervasive.SQL V8 ▸ Documentation ▸ Pervasive.SQL V8.

Printed versions of the following titles are available for purchase:

- *Getting Started with Pervasive.SQL* (Server or Workgroup edition)
- *Pervasive.SQL User's Guide*
- *SQL Engine Reference*
- *Status Codes and Messages*
- *Pervasive Products and Services*
- *Advanced Operations Guide*
- *What's New in Pervasive.SQL (for latest release)*

To order manuals, please contact Pervasive Software using one of the following methods:

Online through the Pervasive Store: http://www.pervasive.com/commerce

E-mail: salessupport@pervasive.com

Phone: 1-800-287-4383

Developer Center

The Pervasive Software Web site is a great source for Pervasive.SQL information: http://www.pervasive.com. It is your most immediate source for assistance with the product.

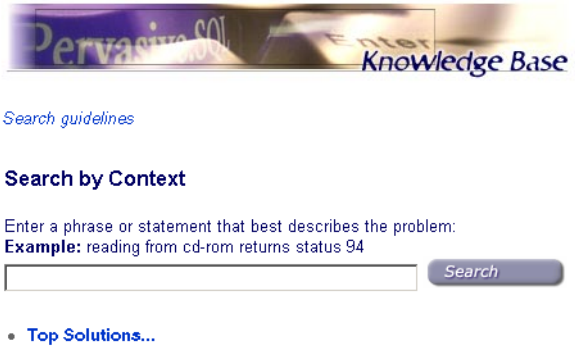
The link shown below is commonly referred to as Developer Center. It is a great starting point from which to navigate to available downloads, documentation, product updates, news articles, sample code and tutorials. Developer Center also provides access to an expansive technical library and training information.

http://www.pervasive.com/developerzone/

Pervasive.SQL Knowledge Base

The Pervasive.SQL Knowledge Base Online is a searchable database for technical information regarding installation, configuration, component management, product defect status, and answers to the frequently asked questions (FAQs). The Knowledge Base, shown below, uses an associative problem-solving technology to perform contextual searches and can be used to quickly find specific answers to your questions about Pervasive products.

http://support.pervasive.com/eSupport/



FTP Site

Pervasive Software strives to maintain close ties to developers using Pervasive.SQL for their database applications. On the Pervasive FTP site, you can find practical resources such as downloadable updates and patches to our product offerings as well as additional debugging tools, documentation, third-party tools, and beta releases.

ftp://ftp.pervasive.com/support/

Online Documentation

The latest versions of Pervasive.SQL product manuals in CHM, HLP and PDF format are available for download from the Pervasive Software web site. These titles include:

- *Getting Started with Pervasive.SQL* (Server or Workgroup edition)
- *What's New in Pervasive.SQL (for current release)*
- *Pervasive.SQL User's Guide*
- *SQL Engine Reference*
- *Status Codes and Messages*
- *Pervasive Products and Services*
- *Advanced Operations Guide*

http://www.pervasive.com/support/technical/online_manuals.asp

Pervasive Library

You can also view all the current documentation online, along with technical papers, discussion forums, and other resources.

http://www.pervasive.com/library

Webinars

Pervasive Software offers a series of Webinars (web-based seminars) that are focused on singular subjects and have been designed to maximize the transfer of knowledge to our partners and developers without expense of travel.

This forum also offers opportunities for live contact with subject matter experts from Pervasive Software in lively discussions during informal and open question-and-answer periods provided throughout the conferences.

For more information, see http://www.pervasive.com/training/calendar/index.asp.

Subscription Based E-mail Services

TechWire and NewsWire are newsletters on all things Pervasive: learn about Beta cycles and releases, Service Pack releases, current topics, FYIs, FAQs, Pervasive Software Events in your area, trade shows where you can find us, and much more!

To subscribe to these free e-mail services, see the following website.

http://www.pervasive.com/support/subscription.asp

DevTalk

Pervasive Software's DevTalk discussion forums are a great way to share ideas with other customers, get technical questions answered, and give feedback directly to Pervasive Software.

http://www.pervasive.com/devtalk/

Newsgroup

Many Pervasive.SQL customers enjoy participation in a newsgroup—a learning environment in which users help users, with some participation by Pervasive Software. The newsgroup is managed by the end-user community, posting and answering questions as they wish.

Pervasive Software is represented in the worldwide network of news discussion groups at:

news://comp.databases.btrieve.

E-Mail Contacts

Pervasive Software welcomes your comments, suggestions and requests for assistance via e-mail. Please submit to the following contacts:

- docs@pervasive.com

  For comments or concerns regarding the content of Pervasive.SQL documentation. Requests for manuals should be directed to your sales rep or "salessupport@pervasive.com".

- uteam@pervasive.com

  For comments or concerns regarding the Pervasive.SQL utilities, such as PCC, PSA, installation, and other Pervasive tools.

- salessupport@pervasive.com

  For information about Pervasive.SQL sales matters such as contacts, pricing, and product specifications.

- developer@pervasive.com

  For developer relations. A great way for developers to communicate their ideas about all Pervasive products, interfaces and programs.

- info@pervasive.com

  For general information about the company, marketing efforts, public relations, and other general questions.

- investor.relations@pervasive.com

  For questions from investors.

Technical Support

You may obtain technical support from the following Web-based support options:

- Pervasive Knowledge Base at http://support.pervasive.com/eSupport
- To open a support incident, http://www.pervasive.com/support/index.asp. Click on **Open an Electronic Incident**.
- To report a product defect, http://www.pervasive.com/support/index.asp. Click on **Report a product defect**.

If you require something other than what Web-based support options provide, contact Pervasive Support by phone:

- 800-287-4383 option 3 (the Americas)
- CIC: +00.800.1212.3434 (Belgium, France, Germany, Italy, Luxembourg, The Netherlands, Spain, Sweden, Switzerland, and the U.K)
- CIC: +32.0.23.37.61 (Any other European, Middle Eastern, African or Asian countries, excluding Japan)

For technical support and discussions about Pervasive products in general:

- Visit the DevTalk forum at http://www.pervasive.com/devtalk.
- Visit http://www.pervasive.com/company/contact/index.asp for other contact information.

Index

Symbols

.PVS file 5-7

A

Access conflict 9-18
Access methods 9-26
Access rights
  adding a group 3-23
  adding a user 3-25
  cannot perform the operation 9-8
  problems due to lack of O/S rights 9-4
  See also Rights
Accessing
  remote database 2-43
Act as part of the operating system 2-14
Active Directory
  create Pervasive_Admin group on domain controller 2-14, 2-16
Active grid
  See SQL Data Manager
Add column button 3-16
Add index button 3-17
Add segment button 3-17
Adding
  databases 3-10
    See also Table Designer
  groups 3-23
  tables 3-15
    See Table Designer
  users 3-25
Admin share not supported 9-19
Administrative rights 2-10
  for Active Directory 2-11
  granting 2-10
  required to create a database 3-10
  troubleshooting 9-17
  user rights policy 2-14
Administrator user
  fast user switching 2-56
ADO 1-11
Advanced Operations Guide 1-18
Advantages of Pervasive.SQL 1-10
Analyzing
  your system 7-1
Attributes
  column 3-17
Autostart
  keeping Workgroup engine from autostarting 9-16
Available sessions
  error in PCC if none available 9-9

B

Backup
  online 9-31
Benefits
  of Pervasive.SQL 1-10
Bound databases 2-55
BSTART, loading NSS volumes first 2-8
Btadmin utility 8-6
Bti.ini 2-37, 2-38, 8-26
Btrieve
  and terminology 1-9
  owner names 9-25
  testing 7-1
  version 6.x and Pervasive.SQL 9-16
Btrieve.nlm
  mixing versions of NLMs not permitted 9-27
  nwmkde.nlm and 9-27
  SQL engine and 9-27
BTRV UNLINK 2-9
BUTIL
  Linux and 8-8

C

Cannot access data after upgrade 9-19
Cell, defined 1-6
Changing
  hard coded file path in DDF 9-21
  order of columns 5-40

Check Database Wizard
name of .exe 3-8
Checkdb.exe 3-8
Checking
database consistency 3-43
Client
definition of 1-4
DSN 2-22
defined 2-21
DSN setup 2-43, 2-45
installation 9-15
setting up database access 2-40
troubleshooting network protocols 9-6
Clilcadm utility 8-17
Clustering 9-15
CNVDDF utility 9-21
Coexistence with previous versions 9-16
Column
attributes 3-17
changing order of 5-40
defined 1-5
deleting 5-41
inserting 5-39
Microsoft Access limited to 256 columns 2-50
set case sensitivity 5-45
set collating sequence 5-46
set data type 5-42
set default 5-47
set name 5-41
set precision 5-43
set scale 5-44
set size 5-42
set to allow or disallow nulls 5-44
too few when creating table for existing data 9-28
Commenting SQL statements 6-9
Compatibility
between Server and Workgroup 1-12
between versions
where to find information 9-29
Components
of Pervasive.SQL 1-3
Concepts
basic database structures and terms 1-5
databases 1-2, 1-5
ODBC standard 2-21
Configuration
bti.ini 2-37, 2-38, 8-26
delay in, eliminating 3-5, 3-6
engine DSN 2-37
odbc.ini 2-38, 8-27
server
O/S rights required 9-4
Conflict, access 9-18
Connection strings
'ServerDSN' or 'DBQ' not found in 9-2
Consistency checking 3-43
Contacting
Pervasive resources 10-1
Continuous Operations
behavior after server failure 9-31
Converting
data files to older file format 9-20
v3.01 DDFs to current version 9-21
Cost of ownership, lowest 1-10
Create Database Wizard
creating a Client DSN 2-40
creating an Engine DSN on NetWare 2-31
creating an Engine DSN on Windows 2-25
name of .exe 3-8
prerequisites to creating a database 2-24
CREATE GROUP 9-25
CREATE PROCEDURE
error code -4994 9-11
Create Table Wizard
column attributes 3-17
name of .exe 3-8
CREATE USER 9-25
Createdb.exe 3-8
Creating
databases 3-10
O/S rights required 9-4
See Table Designer
DDFs for existing data 9-23
DDFs for old data, record size is wrong 9-20
DSN on NetWare 9-24
table for existing data file, too few columns 9-28
tables 3-15
tables See also Table Designer
users 3-25

Creating an SQL query with SQL Data Manager 6-44
Cross-platform support 1-10
Crtblwzd.exe 3-8

D

Data
  cannot access, troubleshooting 9-3
  exporting 3-33
  importing 3-33
  modifying 3-32
  stored in files 9-26
  viewing 3-32
Data definitions
  stored in files 9-26
Data Dictionary Files *See* DDFs
Data Export Wizard 3-33
  name of .exe 3-8
Data files
  changing location of 5-53
  converting to older file format 9-20
  not affected by install/uninstall 9-15
  re-use of space 9-30
  similar
    sharing of DDFs 9-25
Data Import Wizard 3-33
  name of .exe 3-8
Data replication 9-31
Data Source Names
  about 2-22
Database
  access
    prerequisites 2-24
    setting up 2-21
    setting up on client 2-40
  binding 2-55
  concepts 1-2
  consistency checking 3-42
  defined 1-2, 1-6
  engine
    administrative rights 2-10
    definition of 1-3
    starting 2-2
      with PCC 3-28
    stopping 2-2
      with PCC 3-28
  inconsistencies, check database wizard and 3-46
  mirroring 9-31
  named 2-22
  securing a 3-22
  security
    logging in as an administrator 2-20
  security vs OS security 9-16
  structures 1-5
Database administrator, not required 1-10
Database Management System
  defined 1-2
  functions of 1-3
Database names, cannot retrieve (error) 9-8
Database Server Engine
  starting and stopping
    Windows NT 2-3
Database Workgroup Engine
  starting and stopping
    Windows 2-7
Databases
  adding 3-10
  creating 3-10
  deleting 3-13
  removing 3-13
dbmaint 2-37
Dbmaint utility 2-37, 8-18
DBMS, see Database Management System
DBNAMES (Named Databases) 2-22
DBQ
  not found in connection string 9-2
DDF Builder 9-23
DDF Ease 9-23
DDF Sniffer 9-23
DDFs 2-55
  building for existing data 9-23
  changing hard-coded file path 9-21
  converting from old version 9-21
  detecting non-standard 9-22
  how to keep safe 9-22
  mixing among databases 9-23
  must exist for ODBC access 2-24
  sharing among data files 9-25
  verifying against data files 3-43

Definitions
cell 1-6
client 1-4
column 1-5
database 1-2, 1-6
database engine 1-3
engine 1-3
field 1-5
index 1-6
join 1-8
local 1-7
Namespace 3-2
record 1-5
remote 1-7
requester 1-4
row 1-5
schema 1-6
table 1-6
value 1-5
Delay
in Configuration, eliminating 3-5, 3-6
Delete Database Wizard 3-13
Deleted records, re-use of disk space 9-30
Deleting
a table 3-20
databases 3-13
See Table Designer
DSN 2-52
group 3-26
user 3-26
Detecting
non-standard DDFs 9-22
Determining
if DDFs are non-standard 9-22
if ODBC interface is used 9-21
network protocol in use 9-18
record length 5-38
service pack level 9-15
DevTalk, web forum 10-9
Directory field
in Create Database Wizard 2-27
Disk space
re-use of 9-30
Documentation 1-17
Advanced Operations Guide 1-18
different formats 9-17
Getting Started 1-17
Online help 1-19
Pervasive Products and Services 1-18
SQL Engine Reference 1-18
Status Codes and Messages 1-18
User's Guide 1-18
What's New 1-18
Drop Database Wizard
name of .exe 3-8
Drop Table Wizard
name of .exe 3-8
Dropdb.exe 3-8
Dropping
a Table
See also Table Designer
a table 3-20
Droptab.exe 3-8
DSN
client 2-43
defined 2-21
Linux client 2-45
client DSN needed for many applications 9-6
concepts 2-21
creating on NetWare 9-24
engine
defined 2-21
on a Linux server 2-37
File DSN not supported 2-21, 2-43
recreating 2-23, 2-52
removing 2-52
Dsnadd utility 2-45, 8-20
Dump file
cannot read it 9-29

E

Elements
of Pervasive.SQL 1-3
E-mail
Pervasive newsletter 10-8
Enforcing Referential Integrity 2-55

Engine
components
different version than client 9-2
defined 1-3
DSN 2-22
defined 2-21
stopping Workgroup under fast user switching 2-57
tray icon with fast user switching 2-57
Error
'ServerDSN' or 'DBQ' not found 9-2
-4994 9-11
error 1114 9-2
Error was encountered message
troubleshooting 9-10
Excel
pasting data from SQL Data Manager 6-60
Existing data file
creating DDFs for 9-23
too few columns when creating table for 9-28
Export files
cannot read file 9-29
Export Wizard *See* Data Export Wizard
Exporting data 3-33
expwizrd.exe 3-8

F

Failover 9-15
FAQ, see Frequently asked questions
Fast user switching 2-56
running Workgroup engine 2-57
tray icon with 2-57
types of users for 2-56
Features
of Pervasive.SQL 1-10
Field, defined 1-5
Field.ddf 9-22
File
format
converting data to older 9-20
version upgrade not needed for SQL access 9-27
path
changing hard-coded in DDF 9-21
where data and metadata stored 9-26
File DSN not supported 2-21, 2-43
File system security 1-20, 2-46
File.ddf 9-22
Files
explanation of required 9-26
Finding
information on upgrading 9-29
non-standard DDFs 9-22
service pack level 9-15
Foreign key
modifying table definition and 5-6
Frequently asked questions 9-12

G

Garbage collection
on disk 9-30
Getting Started guide 1-17
GRANT 9-25
Granting
administrative rights 2-10
for "this account" 2-13
on Linux 2-19
on NetWare 4.2, 5.x or 6.x 2-19
on Windows 2000 2-12
on Windows NT 2-12
Grid Results
See SQL Data Manager
Group
adding a user to 3-27
adding to database 3-23
deleting 3-26

H

How
to access data via ODBC from other applications 2-46
to add
existing user to a group 3-27
new user to the database 3-25
to apply a user count upgrade 9-18
to check
for orphan rows 3-44
referential integrity 3-44
whether table definitions match data file structure 3-43

to create
new database 3-10
new table in a database 3-15
table definition for an existing data file 3-19
to delete
existing database 3-13
existing table from a database 3-20
existing user 3-26
to display table properties 3-29
to export data from an existing table 3-33
to find
information 1-17
to grant a user administrative rights
for "this account" 2-13
on Linux 2-19
on NetWare 4.2, 5.x or 6.x 2-19
on Windows 2000 2-12
on Windows NT 2-12
to import data into an existing table 3-36
to interpret engine status icons in PCC 3-6
to list referential constraints 3-42
to login as administrator 2-20
to obtain more help 10-1
to register a remote server in PCC 3-4
to remove a remote server from PCC 3-4, 3-5
to set up a client DSN
on a Linux workstation 2-45
using ODBC Administrator 2-43
to set up database access
for a client workstation 2-40
on a Linux server 2-37
on a NetWare server 2-31
on a Windows server 2-25
on a Workstation or Workgroup engine 2-25
to start and stop database engine
on Linux 2-9
on NetWare 2-8
on Windows server 2-3, 2-6, 2-7
to start and stop Windows services within PCC 3-28
to start database engine by using Net Start or Btisc.exe
on Windows server 2-6
to stop database engine by using Net Stop or Btisc.exe
on Windows server 2-6
to turn off database security 3-23
to turn on
database security 3-22
to verify database consistency 3-44
to view
registered database engines 3-6

I

Icons
in PCC, interpreting 3-6
Identifying
service pack level 9-15
Import Wizard, see Data Import Wizard
Importing data 3-33
impwizrd.exe 3-8
Inconsistencies, database
repairing 3-46
Increasing user count licenses 9-18
Index
defined 1-6
Index.ddf 9-22
Indexes
See Table Designer
Information
how to find 1-17
Inserting
a column 5-39
Installation
client FAQ 9-15
data files not affected by 9-15
reboot NetWare 5.x after 9-16
InstallScout
replaced by Pervasive System Analyzer 7-1
Integrity enforced 2-55
Interfaces
ADO 1-11
ODBC 1-11
OLE-DB 1-11
supported for programming 9-26

J

Join
defined 1-8
Joining tables in SQL Data Manager 6-13, 6-87

L

Legacy data
  creating DDFs for 9-20
Legacy file format
  converting data files to 9-20
License Administrator 4-1
  CLI tasks (how to)
    apply license key 4-16
    display all applied licenses 4-16
    display help 4-17
    display specific license 4-16
    remove a license 4-16
  command line interface 4-2
  graphical user interface 4-2
  GUI tasks (how to)
    apply license key 4-13
    determine total user count 4-15
    display applied license 4-14
    display help 4-15
    remove a license 4-14
    select a computer for license administration 4-11
    start as stand-alone application 4-11
    start from PCC 4-10
  GUI visual reference 4-5
  license key 4-2
  licenses for prior releases 4-4
  tasks (how to) 4-9
  user count 4-2, 4-3
    applying 4-4
    obtaining 4-4
License count
  increasing 9-18
  See User count
License key
  administering licenses with License Administrator 4-1
  and user count 4-2, 4-3
  applying a key 4-13, 4-16
  applying user count 4-4
  determining total user count 4-15
  displaying a specific license 4-16
  displaying all applied licenses 4-14, 4-16
  for prior releases 4-4
  obtaining user count 4-4
  removing a license 4-14, 4-16
  starting License Administrator 4-10, 4-11
Licensing
  workgroup engine 9-18
Limited user
  fast user switching 2-56
Linked mode
  See Table Designer
Linux
  client DSNs 2-45
  configuration
    bti.ini 2-37, 8-26
  engine DSNs 2-37
  manual pages 8-4
  MicroKernel Database Engine, starting and stopping 2-9
  naming databases 2-37
  user manual exclusions 8-2
    changes/exclusions 8-2
    utilities 8-2
  utilities 8-5
    btadmin 8-6
    butil 8-8
    clilcadm 8-17
    dbmaint 2-37, 8-18
    dsnadd 2-45, 8-20
    mkded 8-22
    Mkded, console mode 8-25
    mkded, daemon mode 8-24
    sqlmgr 8-26
    sqlmgr, bti.ini 2-37, 2-38, 8-26
    sqlmgr, console mode 8-28
    sqlmgr, daemon mode 8-27
    sqlmgr, odbc.ini 2-38, 8-27
Load balancing 9-16
Loading
  database engine 2-2
  NSS volumes before BSTART/MGRSTART 2-8
Local, defined 1-7
Location
  changing for table data file 5-53
Log on authority for service 2-13
Log on failed
  troubleshooting 9-17
Logical AND
  creating with Query Builder 6-67

Login
OS vs DB 9-16
Lowest total cost of ownership 1-10

M

Manual pages 8-4
Map Root not supported 9-19
Maximum
users for workgroup engine 9-18
MGRSTART, loading NSS volumes first 2-8
MicroKernel Database Engine
defined 1-3
starting and stopping
Linux 2-9
NetWare 2-8
Microsoft
Access
accessing Pervasive.SQL data using 2-48
limited to 256 columns 2-50
Excel
accessing Pervasive.SQL data using 2-46
Migrating
NetWare to Windows server 9-28
where to find information 9-29
Windows server to NetWare 9-28
Mirroring, database 9-31
MKDE *See* MicroKernel Database Engine
Mkded utility 8-22
console mode 8-25
daemon mode 8-24
Modifying
data 3-32
table properties 3-29
See Table Designer
Monitor
O/S rights required 9-4
Multiple
processors
use of 9-29

N

Name field
in Create Database Wizard 2-27
Named databases 2-22
Namespace
defined 3-2
Naming
databases
dbmaint utility 2-37
from a Linux server 2-37
NetWare
btrieve.nlm and SQL engine 9-27
load dependencies 2-9
MicroKernel Database Engine, starting and stopping 2-8
migrating to Windows server 9-28
NSS volumes 2-8
slower on updates 2-8
nwmkde.nlm and btrieve.nlm required 9-27
ODBC and 9-23
reboot 5.x after installing Pervasive.SQL 9-16
unloading Pervasive.SQL 2-9
Network
communications
verifying 9-18
protocol
determining 9-18
troubleshooting 9-6
Network communications
testing 7-1
New utilities 7-1
News
obtaining via e-mail 10-8
Non-standard DDFs
detecting 9-22
NSS volume support 2-8
Nulcnvwz.exe 3-9
Null
modifying table definition and 5-6
restrictions for column 5-6
Null Conversion Wizard
name of .exe 3-9

O

ODBC 1-11
concepts 2-21
interface, determining if used 9-21
odbc.ini 8-27
setting up on NetWare 9-23
Odbc.ini 2-38, 8-27
OLE-DB provider 1-11
Online backup 9-31

Online documentation
format varies by computer 9-17
Online help 1-19
Operating system
security vs DB security 9-16
troubleshooting access rights 9-4
Order of columns
changing 5-40
Orphan rows
checking for 3-44
Overview
Pervasive Control Center 3-2
Owner names
and SQL security 9-25
cannot use PCC to grant permissions for files with owner names 3-23
Ownership, lowest total cost of 1-10

P

Password
troubleshooting wrong 9-10
Performance
NSS volumes slower on updates 2-8
Pervasive Control Center 3-2
accessing a remote server 3-4
adding
databases 3-10
column attributes 3-17
Create Table Wizard 3-15
creating
tables 3-15
creating tables
See Table Designer
Delete Database Wizard 3-13
deleting
databases 3-13
eliminating delay in Configuration 3-5
Export Wizard 3-33
Import Wizard 3-33
interpreting engine icons 3-6
overview 3-2
poll interval 3-5
registering a remote server 3-4
setting database security 3-22
SQL Data Manager 3-32
stopping and restarting services 3-28
turning security on/off and 3-22
viewing and modifying table properties 3-29
See Table Designer
viewing database engines 3-6
wizards available 3-8
Pervasive Products and Services 1-18
Pervasive System Analyzer 7-1
name of .exe 3-9
Pervasive.SQL
advantages 1-10
benefits 1-10
client, defined 2-22
components of 1-3
explained 1-2
features 1-10
Server 1-12
server, defined 2-22
Workgroup 1-12
Pervasive.SQL 7
need to recreate DSNs 2-23
Pervasive.SQL utilities. *See* Utilities.
Pervasive_Admin security group 2-10
use with Active Directory 2-14, 2-16
Platform support 1-10
Preserving DDFs 9-22
Primary key
modifying table definition and 5-6
Programming interfaces supported 9-26
Protocol
determining network 9-18
PSA, see Pervasive System Analyzer
psawizrd.exe 3-9
PVS file 5-7
PVSW.LOG
client and server compatibility 9-2

Q

Query Builder
See SQL Data Manager
Query Builder Diagram
See SQL Data Manager
Query Builder Grid
See SQL Data Manager
Query timeout period 6-51
Questions, frequently asked 9-12

R

Record
  defined 1-5
  length
    determining 5-38
    wrong when creating DDFs for old data 9-20
Recreating DSNs 2-23, 2-52
Redirected mapping not supported 9-19
Referential constraints
  listing 3-42
  verifying 3-44
Referential integrity 2-55
  checking 3-42
  how to check integrity 3-44
Registering a remote engine in PCC 3-4
Relational access
  setting up on NetWare 9-23
Remote
  database access 2-43
  defined 1-7
Remove column button 3-16
Remove index button 3-17
Remove segment button 3-17
Removing
  a column
    See Table Designer
  a remote engine from PCC 3-4
  a table 3-20
    See Table Designer
  databases 3-13
  DSN 2-52
Repairing
  database inconsistencies 3-46
Replication 9-31
Requester
  defined of 1-4
  for SQL Relational Database Engine 9-26
  troubleshooting client 9-4
Rights
  adding a group 3-23
  adding a user 3-25
  administrative 2-10
    required to create a database 3-10
  granting 3-24, 3-25
  granting administrative
    for "this account" 2-13
    on Linux 2-19
    on NetWare 4.2, 5.x or 6.x 2-19
    on Windows 2000 2-12
    on Windows NT 2-12
  if changed from system account 2-13
Row, defined 1-5
Rows
  orphan,checking for 3-44
Running
  in trace mode 9-30

S

Scalable solution 1-10
Scalable SQL
  converting v3.01 DDFs to current version 9-21
Scheduling events 9-26
Schema, defined 1-6
Security
  administrative rights 2-10
  database 3-22
  file system 1-20, 2-46
  logging in as administrator 2-13, 2-20
  OS login vs DB login 9-16
  owner names vs SQL security 9-25
  turning database security off 3-23
  turning database security on 3-22
Server
  DSN not found in connection string 9-2
  registering in PCC 3-4
  troubleshooting network protocols 9-6
Service
  log in authority for "this account" 2-13
  stopping Workgroup engine when using fast user switching 2-57
Service pack
  identifying level 9-15
Services
  starting and stopping with PCC 3-28
Setting
  Database Security 3-22

Setting up
database access
Access to database
setting up 2-21
database access on client 2-40
Shadowing, database 9-31
Shared files, troubleshooting 9-19
SmartScout
replaced by Pervasive System Analyzer 7-1
Smithware 9-23
SMP, see Multiple processors
SQL
CREATE GROUP 9-25
CREATE USER 9-25
executing statements 3-32
GRANT 9-25
SQL Data Manager 3-32
active grid 6-5
aggregate functions 6-33
AVG (average) aggregate function 6-33
commenting lines 6-9
Count aggregate function 6-33
criteria operators
< 6-34
<= 6-34
<> 6-34
= 6-34
> 6-34
>= 6-34
BETWEEN 6-34
IN 6-34
IS NOT NULL 6-34
IS NULL 6-34
LIKE 6-34
NOT BETWEEN 6-34
NOT IN 6-34
NOT LIKE 6-34
getting started with 6-40
group by 6-33
joining two tables 6-13, 6-87
main window GUI reference 6-16
About 6-28
Active Grid 6-26
Add Table command 6-25
Change Query Type command 6-25
Clear command 6-24
Close Database 6-18
Continue 6-24
Copy 6-20
Cut 6-19
Developer Center (Web) 6-27
Exit 6-19
File menu 6-17
Find 6-20
Find Next 6-20
Glossary 6-27
Grid Results tab 6-30
Group By command 6-25
How Do I? 6-26
Insert/Overtype indicator 6-31
Knowledge Base (Web) 6-27
New SQL Query command 6-18
Online Manuals (Web) 6-27
Open Database 6-18
Open SQL Query command 6-18
Page Setup 6-19
Paste 6-20
Print 6-18
Progress indicator 6-30
Properties 6-26
Query Builder 6-25
Query Builder Diagram command 6-21
Query Builder Grid command 6-22
Query Pane 6-29
Query Pane command 6-21
Replace 6-20
Results Grid 6-30
Rows fetched indicator 6-31
Run All SQL Statements command 6-24
Run Current SQL Statement in Grid command 6-23
Run Current SQL Statement in Text command 6-23
Save SQL Query As 6-18
Save SQL Query command 6-18
Search Documentation 6-27
Select All 6-20
Set Font command 6-22
SQL Statements 6-27
Stop command 6-24
Text Results tab 6-30
Title Bar 6-17

Topics 6-27
Undo 6-19
What's This? 6-27
MAX (maximum) aggregate function 6-33
MIN (minimum) aggregate function 6-33
panes 6-8
paste copied data into Excel 6-60
primary work areas 6-2
purpose 6-2
Query Builder 6-9
defined 6-8
logical AND 6-67
mixing use of statements 6-10
SQL statements to build with 6-10
Query Builder Diagram 6-11
behavior when no tables in Diagram 6-12
joining two tables 6-13, 6-87
symbols used in table window 6-11
Query Builder Diagram GUI reference 6-31
Query Builder Grid 6-14
behavior when no tables in Query Builder Diagram 6-15
grid for DELETE statements 6-37
grid for INSERT statements 6-36
grid for SELECT statements 6-32
grid for UPDATE statements 6-35
grids specific to SQL statement types 6-14
Query Builder Grid GUI reference for DELETE 6-37
Column 6-38
Criteria 6-39
Or 6-39
Table 6-38
Query Builder Grid GUI reference for INSERT 6-36
Column 6-37
New Value 6-37
Query Builder Grid GUI reference for SELECT 6-32
Alias 6-32
Column 6-32
Criteria 6-34
Group by 6-33
Or 6-34
Order by 6-33
Result 6-32
Table 6-32
Query Builder Grid GUI reference for UPDATE 6-35
Column 6-35
Criteria 6-36
New Value 6-35
Or 6-36
Table 6-35
Query Pane 6-8
cursor movement within 6-9
records fetched indicator 6-6
restrictions 6-3
results tabs 6-4
Grid Results tab 6-4, 6-5
scrolling 6-6
states of Grid Results 6-5
Text Results tab 6-7
SQL statements to use with 6-15
SUM aggregate function 6-33
tasks
add a Group By column to Query Builder Grid 6-88
add a table to Query Builder Diagram 6-86
build a DELETE statement with Query Builder 6-82
build a SELECT statement with Query Builder 6-63
build an INSERT statement with Query Builder 6-79
build an UPDATE statement with Query Builder 6-76
cancel (undo) typing actions in Query Pane 6-53
cancel an undo action (redo) in Query Pane 6-53
change data with Grid Results grid 6-45
clear results from Text Results tab 6-61
clear the contents of a row(s) on Query Builder Grid 6-87
close a database 6-45
continue running SQL statements that were stopped 6-57
copy text from Grid Results tab 6-60
copy text from Query Pane 6-54

copy text from Text Results tab 6-61
create an SQL query or script 6-44
cut or delete text from Query Pane 6-55
cut or delete text from Text Results tab 6-61
delete row(s) from Grid Results tab 6-59
delete row(s) on Query Builder Grid 6-88
display help 6-48
exit SQL DM 6-49
find text in Query Pane 6-53
identify machine and database currently open 6-43
insert a row into Query Builder Grid 6-87
joint tables in Query Builder Diagram 6-87
layout the printed page 6-55
move the cursor within Query Pane and run the SQL statement to which it moves 6-50
move the cursor within Query Pane without running any SQL statements 6-49
open a database 6-44
open an SQL script 6-48
paste within Query Pane 6-55
print the contents of Query Pane 6-55
remove a table from Query Builder Diagram 6-87
replace text in Query Pane 6-54
run a single SQL statement 6-56
run all SQL statements 6-57
save an SQL query as a script 6-47
select text in Query Pane 6-54
select text on Text Results tab 6-61
set default behavior for Grid Results grid 6-59
set Grid Results grid to allow changes to database data 6-58
set query timeout period 6-51
set query type for Query Builder 6-62
set SQL statement separator 6-51
set the font for grid results or text results 6-57
set the font for Query Pane 6-56
show or hide Query Builder Diagram 6-62
show or hide Query Builder Grid 6-62
show or hide Query Pane 6-49
start SQL DM as a stand-alone application 6-42
start SQL DM from PCC 6-43
stop running SQL statements 6-57
toggle insert/overtype mode 6-56
type a comment into Query Pane 6-52
work with grid results 6-58
work with text results 6-61
wrap text within Query Pane 6-52
SQL DM
See SQL Data Manager
SQL Engine Reference 1-18
SQL query
creating with SQL Data Manager 6-44
SQL Relational Database Engine
defined 1-4
early file formats and 9-27
scheduler 9-26
setting up access on NetWare 9-23
SQL requester 9-26
SQL statement timeout period 6-51
SQL statements
commenting lines 6-9
use with SQL Data Manager 6-15
Sqlmgr utility 2-37, 8-26
bti.ini 2-37, 2-38, 8-26
console mode 8-28
daemon mode 8-27
odbc.ini 2-38, 8-27
Start up
keeping Workgroup engine from autostarting 9-16
Starting
database engine 2-2
services with PCC 3-28
Status Codes and Messages 1-18
Stopping
database engine 2-2
services with PCC 3-28
Workgroup engine under fast user switching 2-57
Stored procedures
error code -4994 when creating 9-11
Support
cross-platform 1-10
Switching, fast user 2-56
Symmetric Multiprocessing, see Multiple processors
System account rights 2-13

T

Table
- changing location of data file 5-53
- column attributes 3-17
- creating 5-36
  - See Table Designer
- defined 1-6
- deleting 3-20, 5-36
- displaying information about 5-54
- dropping 3-20
- properties
  - modifying 3-29
  - viewing 3-29
- schema
  - verifying against data file structure 3-43
- viewing data from Table Designer 5-37

Table Designer
- .pvs file 5-7
- data types 5-6
- graphical user interface 5-2, 5-8
  - Btrieve tab 5-25
  - columns tab 5-18
  - indexes tab 5-20
  - main window 5-8
  - statistics tab 5-21
- linked mode 5-5
- modes 5-5
- null restrictions for column 5-6
- purpose of 5-2
- tabbed dialogs 5-3
  - Btrieve tab 5-4, 5-25
  - columns tab 5-4, 5-18
  - indexes tab 5-4, 5-20
  - statistics tab 5-4, 5-21
- tasks
  - access help 5-33
  - allow duplicates in index 5-51
  - change Btrieve column position 5-60
  - change column order 5-40
  - change location of table data file 5-53
  - change mode 5-32
  - create a database 5-34
  - create a table 5-36
  - delete a database 5-35, 5-36
  - delete Btrieve column 5-56
  - delete column 5-41
  - delete index 5-49
  - delete index segment 5-50
  - determine record length 5-38
  - discard all changes 5-37
  - display table information 5-54
  - display table statistics 5-32
  - identify active database and table 5-31
  - identify changes that have not been saved 5-37
  - insert Btrieve column 5-54, 5-55
  - insert column 5-39
  - insert index 5-48
  - insert index segment 5-49
  - open a database 5-33
  - open a table 5-35
  - save changes 5-37
  - set Btrieve column case sensitivity 5-60
  - set Btrieve column decimal places 5-58
  - set Btrieve column name 5-56
  - set Btrieve column size 5-58
  - set Btrieve column to allow nulls 5-59
  - set Btrieve column type 5-57
  - set column case sensitivity 5-45
  - set column collating sequence 5-46
  - set column data type 5-42
  - set column default 5-47
  - set column name 5-41
  - set column precision 5-43
  - set column scale 5-44
  - set column size 5-42
  - set column to allow or disallow nulls 5-44
  - specify index as modifiable 5-52
  - specify index sort column 5-51
  - specify index sort sequence 5-53
  - start as stand-alone application 5-30
  - start from PCC 5-31
  - view table data 5-37
  - work with Btrieve data 5-32
  - work with columns 5-32
  - work with indexes 5-32
- unlinked mode 5-5

Tables
adding 3-15
See Table Designer
changing
See Table Designer
creating 3-15
See Table Designer
deleting
See Table Designer
Terminal Server 9-15
Terminology
Btrieve usage 1-9
cell 1-6
client 1-4
column 1-5
database 1-6
database engine 1-3
engine 1-3
field 1-5
index 1-6
join 1-8
local 1-7
record 1-5
remote 1-7
requester 1-4
row 1-5
schema 1-6
table 1-6
value 1-5
Testing
your system 7-1
Text Results
See SQL Data Manager
Timeout period for an SQL query 6-51
Trace
how to run in trace mode 9-30
Tray icon
with fast user switching 2-57
Troubleshooting
ruling out possible causes
client DSN not available 9-6
disabled client requester 9-4
network outage 9-3
no server (engine) DSN available 9-5
OS permissions problems 9-4
server not accepting requests 9-5
server not running, not installed 9-4
wrong network protocol settings 9-6

U

Unable to connect error
troubleshooting 9-9
Unauthorized access
to data file with owner name 9-25
Understanding
databases 1-2, 1-5
Uninstalling
data files not affected 9-15
Unix *See* Linux
Unlinked mode
See Table Designer
Unloading
database engine 2-2
Pervasive.SQL on NetWare 2-9
Updates
slower on NetWare NSS volumes 2-8
Upgrading
can't get to data after 9-19
user count 9-18
where to find information 9-29
Use advanced settings field
in Create Database Wizard 2-27
User
adding to a group 3-27
adding to database 3-25
deleting 3-26
User count
administering with License Administrator 4-1
applying 4-4
determining total 4-15
how to apply 9-18
license key 4-2, 4-3
obtaining 4-4
troubleshooting expired license 9-9
workgroup engine 9-18
User name
OS vs DB 9-16
troubleshooting wrong 9-10
User rights policy
act as part of the operating system 2-14
for "this account" 2-14
User switching 2-56

User's Guide 1-18
Users
files with owner names and 3-23
types of for fast user switching 2-56
Users namespace node
cannot use with files that have owner names 3-23
Utilities
btadmin 8-6
butil 8-8
clilcadm 8-17
dbmaint 2-37, 8-18
dsnadd 2-45, 8-20
for Linux 8-5
License Administrator 4-1
mkded 8-22
console mode 8-25
daemon mode 8-24
new 7-1
overview 1-1, 2-1, 9-1
sqlmgr 8-26
bti.ini 2-37, 2-38, 8-26
console mode 8-28
daemon mode 8-27
odbc.ini 2-38, 8-27

V

Value
defined 1-5
Version
new, data files not affected by installing 9-15
old, data files not affected by uninstalling 9-15
Viewing
data 3-32
table properties 3-29
Viewing and modifying table properties 3-29
See Table Designer
Viewing database engines in PCC 3-6

W

Web, scalable to 1-10
What's New 1-18
Windows
Database Server Engine, starting and stopping 2-3
Database Workgroup Engine, starting and stopping 2-7
XP operating system
fast user switching 2-56
running Workgroup engine with fast user switching 2-57
support for 2-56
tray icon with fast user switching 2-57
types of users for fast user switching 2-56
Windows server
migrating to NetWare 9-28
Wizards
check database
name of .exe 3-8
create database
name of .exe 3-8
create table
name of .exe 3-8
data export
name of .exe 3-8
data import
name of .exe 3-8
drop database
name of .exe 3-8
drop table
name of .exe 3-8
null conversion
name of .exe 3-9
Pervasive System Analyzer
name of .exe 3-9
Wizards in PCC 3-8
Workgroup engine
database access 2-43
keeping from autostarting 9-16
licensing 9-18
maximum users 9-18
simultaneous access 9-18
stopping under fast user switching 2-57

X

XP
see Windows XP operating system
Xtrieve
replacement for 9-27

Z

Z-DBA 1-10